FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Altitude control of an underwater vehicle based on computer vision

Pedro Miguel Flores Rodrigues

MASTER'S DEGREE IN ELECTRICAL AND COMPUTERS ENGINEERING

Supervisor: Nuno Alexandre Cruz (FEUP)

Co-Supervisor: Andry Pinto (INESC-TEC)

July 31, 2018

© Pedro Rodrigues, 2018

Resumo

A contínua necessidade de melhorar o processo de supervisionamento dos oceanos tem vindo a promover a criação e desenvolvimento de novas tecnologias focadas nesta área. Visto que as missões subaquáticas, por vezes, implicam ambientes hostis e perigosos para a intervenção humana, é comum recorrer a sistemas baseados em veículos subaquáticos para a realização destas missões. No decorrer da extração de informação numa missão envolvendo veículos subaquáticos, a distância entre o veículo e o fundo do mar ou outros obstáculos deve ser cautelosamente controlada para garantir a segurança do veículo e o sucesso da missão. Tipicamente, para lidar com esta tarefa, é utilizado um sistema baseado na tecnologia sonar. Embora esta solução simplifique o problema e seja eficaz na maioria dos casos, pode implicar algumas desvantagens em algumas condições subaquáticas e em alguns veículos com certas especificações.

Nesta dissertação é apresentado um sistema capaz de controlar a altitude de um veículo aquático recorrendo a visão computacional. O sistema foi dividido em 2 módulos, um módulo capaz de calcular a altitude baseado em visão computacional (*Sensor module*) e um módulo que para além de aplicar um filtro de Kalman às medidas, é responsável por controlar este valor (*Filtering and Control module*). O veículo utilizado para validar e testar os módulos criados foi um profiler desenvolvido em [1], cujo principal objetivo é a navegação vertical.

O *Sensor module* foi implementado baseado na utilização de 2 dispositivos lasers colocados paralelamente a uma câmara CCD. O princípio de triangulação foi usado para calcular a distância aos obstáculos. Este módulo é também capaz de devolver informação acerca da qualidade das medidas adquiridas e aplicar operações aritméticas na informação obtida. Os testes realizados mostraram um erro relativo médio igual a 1 % no intervalo de altitudes de 0 a 2.5 m.

Relativamente ao *Filtering and Control module*, a solução adoptada para o controlo é baseado na comutação de 2 controladores, um controlador de velocidade (baseado num controlador PI), e um controlador de posição (baseado num controlador PD). O controlador de velocidade é usado quando o veículo se encontra longe da referência de posição pretendida, e após a comutação, o controlador de posição atua de modo a completar o movimento. O modelo matemático do veículo foi obtido de modo a dimensionar os controladores. O controlador de posição foi dimensionado de modo a garantir que não haveria *overshoot* e que a comutação seria suave relativamente à atuação dos motores. Um método de afinação dos parâmetros do controlador de velocidade foi criado que permite o seu dimensionamento baseado no custo energético, máxima corrente, e exigências temporais. Os controladores foram validados utilizando o simulink toolbox do Matlab.

Utilizando o simulador de ambientes aquáticos UWsim, a arquitectura desenvolvida para integração dos 2 módulos foi testada. Durante os testes realizados, um movimento descendente foi executado de modo a testar os controladores dimensionados, a comutação entre os controladores, a interação entre os 2 módulos, e a eficácia da fase de filtragem em casos em que erros foram induzidos no *Sensor module*. Os resultados obtidos provaram que o sistema interage corretamente, os controladores induziram uma boa transição em direção à referência de altitude, e a fase de filtragem foi capaz de lidar com os erros induzidos.

Abstract

The desire of improving and developing new technologies targeting the ocean's supervision is continuously increasing. Since underwater tasks might involve hostile environments far too hazardous for human, it is typical to resort to systems based on underwater vehicles. During the extraction of information, the distance between the vehicle and the sea floor or other obstacles must be warily controlled to ensure its safety and the reliability of the missions. Commonly, to deal with this task, a system based on sonar technology is used. Although this solution simplifies the problem and is effective in most cases, it carries disadvantages in some underwater conditions and in some vehicles with certain specifications.

In this work it is presented a system capable of controlling the altitude of an underwater vehicle using computer vision. This system was divided into 2 modules, a module responsible for measuring the distance based on computer vision (Sensor module) and a module that applies a Kalman filter to the data and is capable of using the filtered data to control the altitude of the vehicle (Filtering and Control module). The vehicle used in order to validate and test the modules created was a profiler developed in [1] which main purpose is the navigation in the vertical direction.

The Sensor module was implemented based on two laser pointer devices placed parallel to one another beside a CCD camera. In order to calculate the distance of the vehicle towards the obstacle, the laser triangulation principle was used. Furthermore, the Sensor module is capable of retrieving information about the quality of the measurements and apply mathematical operations like circular average. The tests performed showed an average relative error equal to 1 % in the range from 0 to 2.5 m.

Relative to the Filtering and Control module, the solution adopted regarding the control stands on the switching of two controllers, a velocity controller (based on a PI approach), and a position controller (based on a PD approach). The velocity controller is used when the vehicle is far away from the target position reference, after the switching point the position controller is used in order to reach the reference. The mathematical model of the vehicle dynamics was used in order to design the parameters of the controllers. The position controller was designed in order to achieve a smooth switching between the controllers and also to guarantee no overshoot. A method of designing the velocity controller was created where the parameters can be tuned based on the energetic cost, maximum current, and temporal demands that depend on the mission's demands. The designed controllers were validated using the simulink toolbox from Matlab.

Using the UWsim underwater simulation environment the overall architecture developed integrating both modules was tested. During these tests a descent movement was performed testing the performance of the controllers designed, the switching of the controllers, the interaction between the 2 modules, and the effectiveness of the filtering phase in cases where errors were induced in the Sensor module. The results obtained showed that the overall system is interacting correctly, the controllers allowed a good transition response towards the position reference, and the Kalman filter was able to handle the errors induced.

Agradecimentos

Ao professor Nuno Cruz, um especial agradecimento pela exímia orientação ao longo desta dissertação, pelas criticas, atenção, e mais importante, pelos ensinamentos e conhecimento que me transmitiu.

Quero também agradecer ao professor Andry Pinto pela ajuda ao longo da dissertação e ao João Monteiro que se mostrou sempre disponível a esclarecer qualquer dúvida acerca do profiler ao longo deste caminho.

Não poderia deixar de agradecer a todos os meus amigos mais próximos que de uma forma directa ou indirecta, contribuíram para a elaboração deste projeto, pelo apoio e sobretudo pelos momentos de descontração que proporcionaram.

Quero também agradecer aos meus pais pelo seu apoio incondicional. Um especial agradecimento ao meu avô pela educação durante a minha infância que tanto contribuiu para a forma como encaro os obstáculos ao longo da minha vida.

Pedro Rodrigues

"Sometimes you may be the smartest person in the room and you do not want anybody to know that. And there are other times that you are not the smartest person in the room but you want to make other people think you are"

Michael Franzese

Contents

List of Figures

List of Tables

Abbreviations

AUV	Autonomous Underwater Vehicle
CCD	Charge Coupled Device
FEUP	Faculdade de Engenharia da Universidade do Porto
FIFO	First-In-First-Out
INESC-TEC	Instituto de Engenharia de Sistemas e Computadores - Tecnologia e Ciência
MOS	Metal Oxide Semiconductor
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
RBF	Radial Basis Function
ROV	Remotely Operated Vehicle
SDK	Software Development Kit
UDP	User Datagram Protocol
XML	Extensible Markup Language

Chapter 1

Introduction

The ocean can be seen as Earth's central engine when it comes to maintaining its balance. Not only it contains the vast majority of Earth's biodiversity and an extreme strategic importance to the security of a nation but it also holds tremendous amounts of resources that enrich the daily life of humankind. It's not only important to fully take advantage of the richness that oceans and rivers posses but also to make sure we manage it well, so those resources are around for future generations. In order to accomplish it, we need a deeper understanding of the properties and state of those ecosystems.

In the depths of the world's oceans, there are "gaps in the unknown" that could have a huge profit potential in a vast number of ways, like deep sea mining, oil, and gas. Information gathered from the ocean can be used to better understand the dynamics of our planet, our history, and culture and can reveal new resources and discoveries with economic, health, and quality of life benefits [6].

Nowadays, there are a vast number of ways to gather information from the ocean, like basic observation, distribution of sensors across the sea, and the use of aircraft and satellites to gather images providing information from the surface layer of the ocean. Even though a lot of data can be gathered by these methods, most of the ocean can only be explored via underwater exploration missions.

There are also several needs relative to underwater structure inspection and repair, recovery of lost man-made objects, ocean floor analysis [7], and crucial roles relative to military purposes to be fulfilled, like security patrols[8].

It is usual that these types of underwater tasks involves hostile environments far too hazardous for humans, making the human intervention not possible in some missions. Therefore, there is the need to resort to systems that make it possible to execute these types of missions reliably and efficiently.

1.1 Context

In order to overcome the challenges that the underwater missions bring, a variety of systems have been developed over the years.

One of the solutions is systems based on Remotely Operated Vehicles (ROVs). These vehicles are typically used for sub-sea pipeline and power transmission cable inspections [9, 10] and tasks that demand human supervision and acquisition of real-time information [11]. ROVs can also be used for several other types of missions, such as ocean observation, surveys, drilling, and construction support and object location and recovery. However, the average high cost of an ROV mission and the inherent necessity of a cable connected to the vehicle, allowing the control of its movement by an operator, limits the vehicle viability in some deep-sea missions [12, 9, 13].

Some missions require complete autonomy on the vehicle's movement and navigation. This is accomplished by the use of Autonomous Underwater Vehicles (AUVs), which are capable of autonomous control of their behavior through the acquisition of data carried out by sensors embedded on the vehicle. AUVs present more flexibility and autonomy compared to ROVs and can be more recommended for carrying out some underwater missions based on exploration and inspection since they require less human intervention and have a lower operative cost [9, 12, 10]. In cases of deep ocean exploration, where the control through a human is not ideal, the AUVs perform a specially important role. These vehicles are used in hazardous operations such as recovery of lost man-made objects, oceanographic observations and ocean floor analysis [8].

1.2 Motivation

During the extraction of information, the position control of the vehicle is critical. Specifically, the distance between the vehicle and the sea floor or other obstacles must be warily controlled to ensure its stability and safety. Commonly to deal with this, a system based on sonar technology is used [14, 15, 16]. Although this solution simplifies the problem and is effective in the cases shown, it carries some disadvantages in some underwater conditions [7] and in some vehicles with certain specifications. Particularly the sensors based on acoustic waves, like the sonar, might present difficulties on the interpretation of the signals received when the vehicle is close to the obstacle, requiring a minimum distance to retrieve valuable and reliable information [13, 2]. Furthermore, the inclusion of the sonar sensor demands an increase on the energetic cost of the system that in the case of vehicles powered by an external source through a cable like the ROVs is not a problem, but in AUVs, it might be valuable to avoid it since these vehicles are powered by batteries. Lastly, sometimes the space occupation of the sonar sensor represents a problem in some vehicles with meticulous limits relative to space usage, a common problem found in AUVs.

To overcome the problems mentioned, another approach to the determination of the distance of the underwater vehicle towards an obstacle is its calculation based on computer vision, which is specially convenient since most of the underwater vehicles have already an embedded camera. Since the problems described are more relative to AUVs, the solution based on computer vision

is specially relevant for this type of vehicles, not being inappropriate to the remaining underwater vehicles.

Several solutions have been proposed based on the use of laser ranging systems that prove the reliability and efficiency of the utilization of this approach in the determination of distances to obstacles by the vehicle. The information gathered by this sensing type is based on the capture of images containing known marks that are detected by image processing, allowing the calculation of the distance towards obstacles and other information that can contribute to the control of the navigation.

Additionally, this technology opens a vast number of new features that are possible to implement and would enrich the system like the acquisition of velocity, orientation, and allowing other functions like mapping and infrastructure inspection [17, 18].

1.3 Goals

The main goal of this work is the development of a system capable of controlling the altitude of an underwater vehicle using computer vision. To provide this functionality, it is required that the system gathers information about the altitude of the vehicle in real-time based on capturing and the processing of images containing known marks imposed by laser devices. Through this information, the system must be capable of holding an ordered constant altitude towards the sea floor during the progression of its mission.

The vehicle used in order to validate and test the system created was a profiler developed in [1], illustrated in figure 1.1. The application of the system to be developed is opportune in this vehicle since it has energy limitations due to being an AUV, reduced diameter making it not possible to incorporate a sonar device, and it navigates mainly in the vertical direction, which makes the control of its altitude extremelly important to control. In this work the concept of altitude is interpreted as the distance of the vehicle towards the closest obstacle in its movement direction.

1.4 Document organization

The present document describes the work done during the development of a system able to control the altitude of a profiler using computer vision, the tests performed and the conclusions obtained.

Chapter 2 - Presents a review of a vast number of solutions available in the scientific area that produce the best results for the ambition of this dissertation.

Chapter 3 - Initially, describes the overview of the system developed and its division into 2 modules, as well as the requirements that were considered for each module. Furthermore, the implementation and respective validation for each module are presented.

Chapter 4 - Using the underwater environment simulator UWSim, the integration of the 2 modules in the profiler was simulated in order to validate the architecture, performance of the

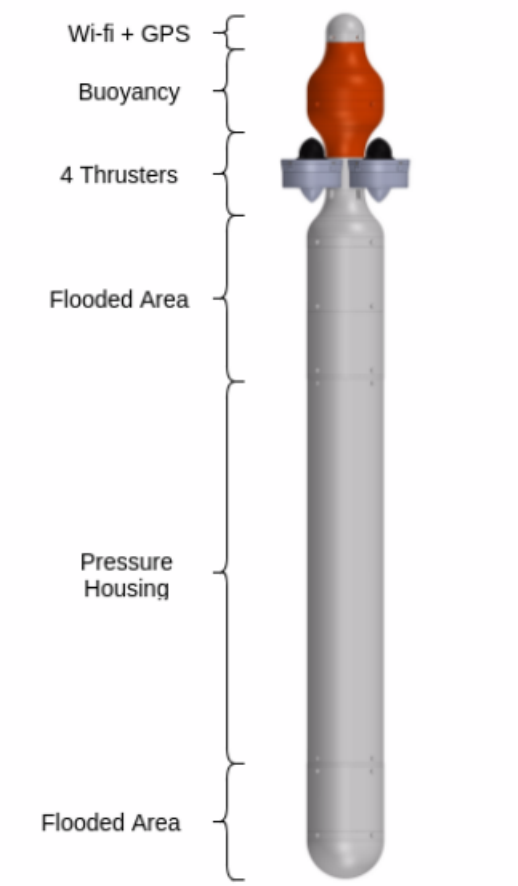

Figure 1.1: Image of the configuration of the profiler

discrete controllers, interaction between the 2 modules, and the performance of the filtering phase. This chapter presents the procedures taken and the results obtained using the UWsim.

Chapter 5 - This chapter concludes the work done, presenting the main contributions and some suggestions for future work to further improve the system developed.

Chapter 2

Bibliographic review

An underwater mission involving an underwater vehicle requires the control of the involved vehicle position and movement. The control of the vehicle can be achieved through human supervision or by autonomous control. ROVs are vehicles controlled by commands, sent through a cable connected to the vehicle, given by an operator. The control is based on the observation and analysis of the state of the vehicle by the operator, using a stream of real-time data gathered by a camera and other embedded sensors that can give relevant information about the vehicle such as distance to obstacles.

On the other hand, AUVs' navigation is accomplished through autonomous control using sensors placed on the vehicle. Since there is no physical human intervention, the viability and efficiency of the control system are highly dependent on the reliability of the sensors, making their selection and the choice of the sensing method demanding challenges of high impact. The sensing method allows the gathering of information in real-time about the vehicle surrounding, making possible the control of its thrusters. This can be accomplished by:

- Acquisition of information about the vehicle surrounds, especially regarding distances to objects that surrounds it, through sensing;
- Filtering and processing of the information gathered and determination of the actuation on the thrusters of the vehicle;
- Execution of the actions selected;

A lot of information regarding the vehicle control can be acquired by different types of sensors. Information like the depth of the vehicle, using pressure sensors [19], orientation via compasses, and distance towards obstacles can be useful on the control of the underwater vehicle. The distance between the vehicle and the obstacles that surround is determinant on the reliability of the navigation, especially in missions that require that the vehicle positions itself extremely close to the sea floor or other objects of interest during its navigation.

This dissertation focal point is the development of a system capable of acquiring the altitude of the underwater vehicle using computer vision and an algorithm to control it, making possible its

maintenance while the vehicle gathers relevant information for the mission. Concerning the development of a system capable of the functionalities described above there is the need to accomplish the following two main tasks:

- Development and implementation of a distance measurement system capable of obtaining the distance from the vehicle to the obstacle based on computer vision;
- Creation of a method capable of controlling the altitude of the underwater vehicle based on the information gathered from the distance measurement system.

The usage of this system depends on the vehicle demands. The main focus of the second task presented is the integration of a control system on AUVs since these type of vehicle inherently require a system that allows their autonomous control. Despite this, any type of underwater vehicle would be able to benefit from it, since it provides viable information for several applications. Although ROVs are controlled manually, this system would be beneficial in a situation where a mode that controls the vehicle autonomously is required, for example for maintaining a distance towards an obstacle.

The purpose of the following review is to study the vast number of solutions available in the scientific area to the tasks presented and to attempt to isolate those that produce the best results for the ambition of this dissertation. The review was divided into 3 subsections: Distance measurement system, Control of underwater vehicles, and Filtering. Initially, it is explored the different methods used in the scientific area in order to acquire the distance in underwater vehicles, as well as the components necessary that make up the device, followed by the analysis of the most relevant image processing algorithm used to extract the distance using computer vision. In the second subsection, some approaches directed at the control of underwater vehicles will be mentioned. The last section present a review of the most used filtering methods in the context of controlling underwater vehicles.

2.1 Distance measurement system

2.1.1 Distance measurement method

The distance of the vehicle towards the sea floor is one of the most important information needed for its control. In order to gather this information, several solutions have been presented through the years. The most common solution is the utilization of the sonar technology to gather the distance of the vehicle to the obstacles that surround it [15, 16]. Although this solution is effective in most cases, as mentioned in section 1.2 it carries a lot of disadvantages in some underwater conditions.

One other option to acquire the distance of the vehicle to the sea floor is the use of methods based on computer vision. In [4] it is presented a method based on geometrical triangulation using a laser beam pointer and a Charge Coupled Device (CCD) camera to capture it. This type of camera device is discussed in 2.1.2. The distance of the vehicle to the objects is calculated through

simple laser triangulation. Using this method, a simple and efficient system was created and tested in air and water, showing a measured maximum error less than 30mm in air and a maximum error of 100mm in water in the range from 450mm to 1200mm. After calibration, in both environments, it was possible to achieve an error of 1mm. The calibration needed is simple and can be easily implemented in the initial setup.

A similar method using two laser points and a CCD camera was used in [7]. In this case, two green laser pointing devices were placed parallel to one another beside the camera. Distances are also calculated by using laser triangulation and processing of the images acquired in order to locate the laser points. An experiment in a tank where the vehicle moved horizontally keeping constant distance revealed good results, for example, the maximum error in accuracy was 40mm and the average error in steady state was 20mm. A sea trial was also realized where the vehicle moved horizontally keeping constant distance and angle of a construction base. The waves and currents in the area of the test were calm and good results were obtained. The maximum error in accuracy was 60mm. These results prove that the introduction of a second laser pointer device enhances the performance by introducing redundancy in the system.

The same sensing specification device was also used in [20]. The measurement method was also based on the relationship of the distance between the reflected laser points and the center of the image captured by the camera. Tests where the AUV maintained the altitude imposed (0.33m in a pool of 1.19m) presented an error of 50mm. In this article, the redundancy gained by using two laser pointer is explicitly used in cases of irregularity of the sea floor. It is considered the worst case scenario, selecting the laser dot that corresponds to the lowest distance that represents the distance towards the obstacle that is the closest to the vehicle.

As stated in [18], experiences and tests regarding the measurements using laser pointer devices revealed two main weakness. Firstly, there is an inherent necessity of detection of the laser points in order to guarantee the reliability of the system. In a situation of contact with irregular obstacles and conditions where the laser points could not be detected by image processing, the system tends to fail. The second main weakness is related to the over-sensitivity towards small and neglected obstacles. In order to overcome the weak points mentioned, in [18] it is presented an approach using one sheet laser beam instead of laser point generators. This method allowed the calculation of the distance towards the obstacles based on triangulation by light-sectioning the obstacle. The system consisted of a CCD camera set with a fixed distance to the sheet laser device that captured the lines on the obstacle projected by the laser. After image processing, it was possible to calculate the distance to the target. Furthermore, this ranging method could be used to acquire the target's shape and roughness as well as 3 dimension mapping. Tank tests were realized to test the ranging system on structure tracing and covering. In these tests, the vehicle is manually navigated to the obstacle and then the system is activated in order to trace the structure moving at 0.1m/sec. Images are captured every 500ms. The vehicle starts tracing with a 0.35m offset and converges to the reference distance in 20 secs. During the horizontal tracing, the distance maintenance showed very accurate results, only showing considerable error when it needed to turn left or right to keep tracing the obstacle. The focus of this article was the comparison between the method based on

sheet laser beam and pinpoint laser. Even though the objective is structure tracing, it revealed that the system implemented can achieve better results in distance-keeping compared to the pinpoint laser ranging system.

In [2] an algorithm able to calculate distance information about the vehicle surroundings is presented using a CCD camera and two laser line generators mounted on top of each other and parallel to the camera's viewing axis instead of two laser pointers. The distance calculation is based on a modified pinhole camera model, providing a direct corresponding relation between the coordinates of the object in the world and the frame. This modified model considers that the focal plane is in front of the camera's aperture instead of behind as the classical pinhole model considers. This allows a better perception of the coordinates acquired since it removes the negative sign presented in the relation between coordinates.

The following images illustrate the difference between the classic pinhole model and the modified pinhole model:

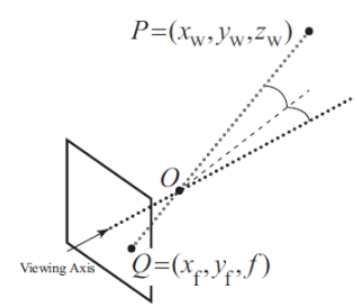

Figure 2.1: Illustration of the classic pinhole model [2]

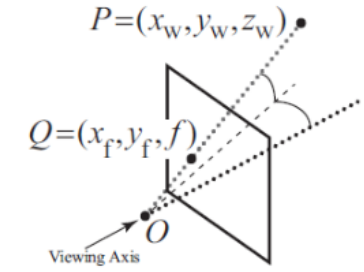

Figure 2.2: Illustration of the modified pinhole model used in [2]

.

Using the classical pinhole mode the relation between the coordinates can be represented by the following equation:

$$\frac{x_w}{z_w} = -\frac{x_f}{f} \quad \frac{y_w}{z_w} = -\frac{y_f}{f} \tag{2.1}$$

The modified pinhole model used in this article allows the following relation:

$$\frac{x_w}{z_w} = \frac{x_f}{f} \quad \frac{y_w}{z_w} = \frac{y_f}{f} \tag{2.2}$$

The distance measurement method is based on the relation of two triangle formed between the camera and the object as illustrated in the following image:

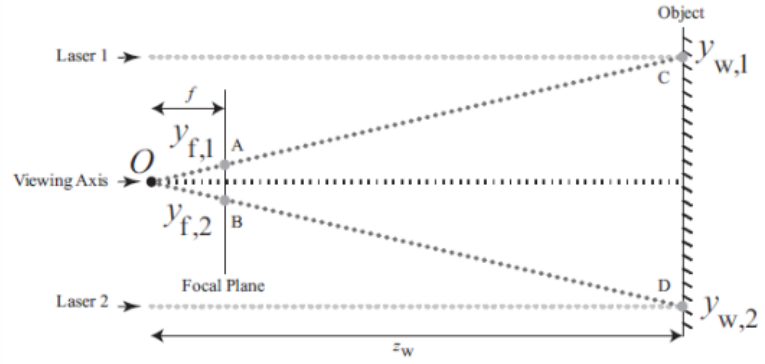

Figure 2.3: Representation of the sensor side view [2]

.

where $\bar{y_w}$ represents the known fixed distance between the laser line generators, $\bar{y_f}$ is the distance between the lines in the image, z_w is the distance between the camera's aperture and the obstacle, and f is the focal length of the camera. Using this method it is possible to calculate the distance to the object represented by z_w by the following relation:

$$\frac{\bar{y_w}}{z_w} = \frac{\bar{y_f}}{f} \tag{2.3}$$

The use of two laser line generators allows the determination of the distance to an object at multiple points of the projected line in the image, enhancing the perception and the performance of the vehicle's ability to navigate. To test the accuracy of the method, an experiment was performed where the results of the sensor were compared with a scanning laser rangefinder as reference. The tests were performed with a distance up to 5m and the approximate absolute error of the method was 10% of the actual distance, which is accurate enough to be used on an underwater vehicle considering its inherent slow dynamics. Since this method uses two laser lines, it is shown that it allows not only the calculation of the distance on multiple locations of the line but also the calculation of the 2D triangulation using only a portion of the image. The reason for this possibility is discussed in 2.1.3. The described advantages reduce the computational complexity and the processing time needed.

2.1.2 Specifications of the distance measurement device

In order to develop a distance measurement system, it is required the choice of the specifications of the measurement device.

Most of the cameras used on underwater vehicles navigation are based on CCD [4, 13, 7, 2, 18]. A CCD is a silicon device composed of a matrix of light sensitive elements, called pixel, which collects photoelectrons when exposed to visible light. Each pixel is a small MOS capacitor that absorbs and captures the photoelectrons in a potential well. After the capture, the stored charges in each element is sent through a serial register and converted into a voltage level. The full extent of each pixel voltage can then be converted to a digital image using an analog-to-digital converter (ADC) [21, 22, 23].

The choice of the laser device, especially the wavelength of its beam is essential to the reliability and efficiency of the sensing algorithm. Since the laser beams reflections must be visible from a significant distance the absorption of the light is a problem that must be considered. In [24], a test in fresh water was done in order to compare the green laser and red laser properties. The red laser shows a power of about ten times the power of the green laser, but its high absorption rate severely impacts its maximum range. Red lasers also presented less scattering when propagating through the water when compared to the green lasers. However, a scattered signal can be filtered and processed to enhance its quality. The test concluded that the green laser was the best for most underwater applications.

In [3] a study of the absorption of visible spectrum of the light in deep ocean, bay and coastal waters was performed, and the results obtained are illustrated in figure 2.4.

Through the observation of the results for deep ocean waters, the coefficient of absorption is the lowest in the spectrum related to the blue and green coloration. Relative to the bay and coastal waters, overall the attenuation is increased through all the spectrum. There is a considerable increase on the attenuation of the blue light compared to the green in bay waters. Considering the overall results, it is clear that the green portion of the visible light spectrum presents the lowest coefficient compared to the rest of the band.

2.1.3 Image processing algorithm

Succeeding the acquisition of the frames from the distance measurement device, it is necessary to process these images in order to provide the information needed to implement the distance measurement methods. Different image processing algorithms are used depending on the conditions that the vehicle is going to navigate, and especially on the specifications of the measurement device used. The image operations used in these algorithms can usually be divided into three main phases: pre-processing; segmentation; analyze and extraction of relevant information [25]. Each operation performed in a certain phase influences the following phases and therefore it is necessary to analyse the different image processing algorithm on its full extent.

In [20] a sequence of image processing operations are used in order to detect and identify the location of the laser points projected. The pre-processing phase starts with the application of a filter to trim the image on the optical axis making not only the processing speed faster but also increasing its robustness. Subsequently, the image is converted to a grey scale and binarized by a method based on a threshold, obtaining a segmented image. The image is then carried out to labeling in order to identify the different laser points. In each of the elements labeled the

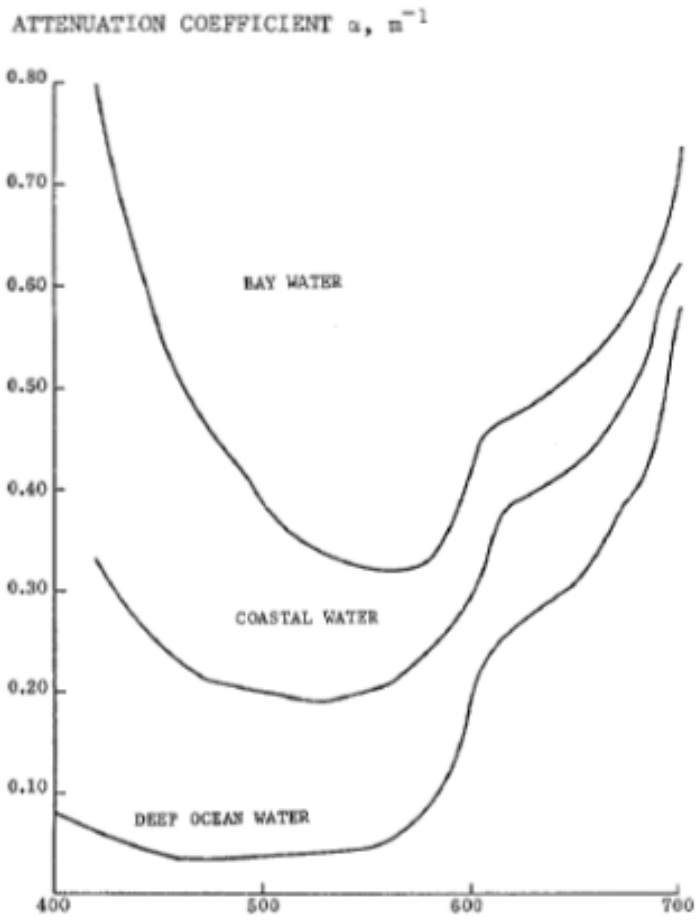

Figure 2.4: Graphical representation of the absorption coefficient of visible light in deep ocean, bay and coastal waters [3]

.

coordinates of the center of mass are acquired and therefore the locations of each laser point are defined.

The algorithm described is considered generic and can be adapted to different applications using a camera and two laser pointers. However, if the method uses laser line generators, the algorithm can diverge considerably. In [2], in order to determine how far the vehicle is from an object, it is necessary to calculate the distance between the two laser lines in the image frame. Initially, the focal point is to remove the distortion applied by the non-ideal behavior of the camera. This is achieved by calibrating the camera in order to match the ideal pinhole camera model. As the algorithm described previously, the next step consists in the segmentation of the image. In this case, since it uses two laser lines instead of two laser points, the segmentation not only allows the calculation of the distance at multiple locations but also reduces the computational effort needed since it grants the possibility of processing a smaller portion of the image. After the image is broken down into smaller segments, it is extracted the location of the laser line in the image using each of the segments. In order to do so, firstly the image is converted into a grey scale one and

then it is applied a threshold in order to get a binary image consisting of only black and white coloration. Since the result of this procedures produces an image with the lines highlighted in white and the rest of the image in black, it is possible to use the Hough Transform to finally extract the laser lines [26]. Each line segment after being extracted are grouped together based on a defined value of pixel separation in order to associate each line to the laser line beam. The last step is to estimate the average middle value of each laser lines. This allows the calculation of the distance to the object based on the distance of the two laser line as described in 2.1.1.

Considering the ranging system used in [18], the image processing procedure is similar to the one used in [2]. The main purpose is to detect and extract the laser lines after reduction of the processing area in order to decrease the sensibility to noise and decrease the processing time.

Multiple tools are available to execute image processing methods like those explained above, with special attention to the OpenCV image-processing library. The algorithms presented in this chapter used this library to implement the image processing operations described [2, 20]. This library allows the implementation of processing techniques and methods like histograms modifications, image enhancement and filtering, morphological operations, segmentation methods, edge detection, Hough transform and advanced image and video analysis and information extraction [27].

2.2 Control of underwater vehicles

Relative to a generic control of an underwater vehicle, it would be necessary to consider the movement of the vehicle in its six degrees of freedom. Since the focus of this dissertation is the altitude control, the focal point is going to be the movement along the vertical plane of the vehicle and therefore the search focus of this study is on the control of 1 degree of freedom.

Although the determination of the control criteria and algorithm depends on the vehicle, the conditions to which it will be subject, the amount and type of sensors available in the vehicle, the objective is the development of a generic underwater vehicle control method capable of controlling 1 degree of freedom of the vehicle.

The most common control approach on position control is the classic Proportional-Integral-Derivative controller (PID), which is a feedback controller. The response of this controller is given by:

$$F(s) = K_p + \frac{K_1}{s} + K_D s E(s) \tag{2.4}$$

This controller operates on the application of a proportional term (P), an integral term (I) and a derivative term, D. The proportional gain contributes to the output based on the existing error, while the integral term increases the action output in relation not only to the current error but also the time for which it has persisted. The integral portion guarantees that if the applied force is not sufficient to bring the error to zero, eventually as time passes it will converge to zero.

The derivative term aims at controlling the rate of change of error in the system, reducing the overshoot[28, 29, 30, 31].

The image presented in 2.5 illustrates the context of the application of this controller in a generic system.

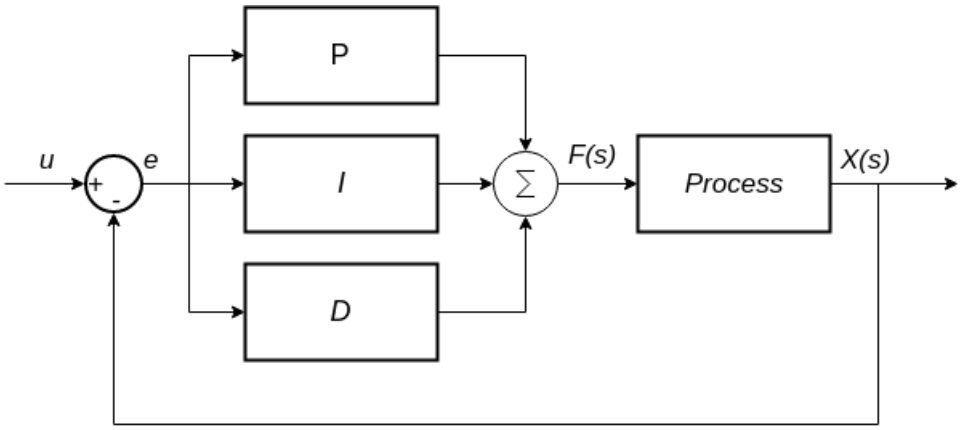

Figure 2.5: Context of application of a PID controller in a generic system

.

In [20] a simple PID controller is presented to control the altitude of the vehicle while it moves horizontally. It was able to converge the altitude value from a 0.5m to a reference of 0.3m in approximately 3s.

Several other cases of success have been reported relative to the control of AUVs based on the PID approach, demonstrating its effectiveness in this ambit [31, 7, 18].

In order to control the movement of a profiler, in [1] a speed controller that controls the velocities in the z axis and a position controller that allows the control of the vehicles' position in the z axis were used. A test where the profiler dived to a defined depth completely vertical was performed. An example of an operation performed was the instruction to dive to 0.4m with a speed of 0.1m/s and the vehicle took around 3s to reach the reference, maintaining this value.

These approaches of control reveals rather underwhelming in more complex tasks where the mathematical model is difficult to obtain precisely. Furthermore, the underwater vehicle's dynamics can be highly non-linear in a marine environment, making these approaches unable to process efficiently, leading to poor performance of the controller [23]. In some cases, the process of linearization of an underwater vehicle is not enough for the pretended performance.

In order to overcome the problems described many other solutions have been proposed, based on adaptive and/or Fuzzy control.

An improved adaptive hybrid fuzzy control for the horizontal motion was used in [32]. Since it is based on fuzzy logic theory, it does not need a precise mathematical model of the vehicle, being based on fuzzy sets and rules articulated through human expertise and experience. It is called adaptive because it allows the tuning of the parameters online based on the performance of the error's change. The main improvements are the addition of an integration factor to the

conventional fuzzy controller in order to eliminate steady-state error, and the output of values with nonlinear rules near the region of zero in order to enhance the sensitivity and precision. This approach presented improvements on the response speed, reduced the oscillation magnitude of static phase, higher resistance and robustness against external disturbance compared to a PID controller, resulting in a higher stability of the control system.

In [33] an adaptive sliding mode with PID tuning method is presented. It uses sliding mode control in order to deal with uncertainty, variation and possible disturbances on the parameters of the process, improving the robustness of the control system. The results demonstrated that the designed controller satisfied the need to deal with the non-linearity and uncertainty when controlling the pitch angle of an AUV.

Another possible approach is the adaptive neural networks control. In [23] and [34] propose adaptive neural sliding methods based on the Radial Basis Function (RBF) neural networks. Simulations and tests realized on those articles demonstrate that an adaptive online learning approach based on RBF neural networks allows the identification of the parameters of AUV's dynamics and induces a better performance compared with the normal adaptive sliding mode.

2.3 Filtering

Besides the control algorithm, the application of filters is a valuable way of removing noise and preparing the information to be given to the control phase, increasing the reliability of the system.

Estimations filter are commonly used to enhance the accuracy in the measurement and to filter out high deviations [35, 13]. The Kalman filter is a general error tracking and estimation based filter. At each time interval, the Kalman filter updates its prediction based on the characteristics of the process, then using the measurements obtained through its sensors, the final value is obtained by adjusting the prediction value with the measurement. The more accurate the measurement the more impact it will have in the correction phase. Furthermore, when the vision system could not acquire the information needed to know the state of the system, it can use the prediction given by the prediction phase or even integrates more information coming through other sensors [36].

In [31], in order to deal with the non-linearity of the model of the underwater vehicle the Extended Kalman Filter was used. In this nonlinear version of the Kalman filter, the model can be fined as differential functions and still retrieve the functionality of estimation of the typical Kalman filter.

As stated in [37] in cases of nonlinear systems, non-Gaussian noise distribution, and extreme irregular measurements, the use of other approaches of filtering other than the Kalman filter, like the particle filter, is recommended. The particle filter is the approximated Bayesian filtering algorithm based on the Monte Carlo method. The idea of this filter is as follows: It is supposed that the state distribution is known and it is also known its probability density function. Extract N samples in the state space which are called particle and represent the hypotheses. These particles are evaluated using the information gathered by sensors in order to know how likely is that state. After updating each particle, the less likely states are substituted by new samples based on

the most likely existing particles. Over time, the number of particles near the correct state will increase gradually, improving the estimation. In [38], the particle filter is used in order to track an underwater vehicle in a swimming pool. The results concluded that the algorithm achieves real-time and stable moving, and overcomes the shortcomings of a typical Kalman filter.

Chapter 3

System overview and implementation

In this chapter, it will initially be presented an overview of the system developed. It will be listed the requirements that were set for this system. Soon after it will be explained how the system was separated into modules and how they were inserted in the overall vehicle's control system. Posterior to that, the principles, thought process and software architecture used in the implementation of these modules will be presented.

It will be also presented and explained all the work done to implement the features necessary in order to achieve the goal of this system. Furthermore, the methods used to characterize and test the implementation will be described, its results and a critical overview of these results will be presented.

3.1 System design

To achieve a system capable of controlling the altitude of an underwater vehicle using sensing based on computer vision, it was developed a module capable of measuring the distance (Sensor module) and a module able to filter the data gathered by the sensor module and capable of using these data to control the distance of the vehicle to an obstacle (Filtering and Control module). Figure 3.1 illustrates a block diagram that demonstrates an high level view of the context where these modules are inserted and how they interact with each other.

The diagram shows that the Filtering and Control module and Sensor module are inserted in a distance control loop sequence of a vehicle based on the input of a reference that is compared to the value obtained by the Sensor module after filtering. Based on this comparation difference the control module operates in order to converge the distance of the vehicle towards the reference altitude by deciding the actuation that is necessary to give to the thrusters of the vehicle.

The division on two modules allows the implementation of the two main inherent features of the system (sensing and control) independently and therefore enables the introduction or the improvement of features on any module without imposing significant changes on the other module. This is especially important since it is intended that the sensor module is developed in a way

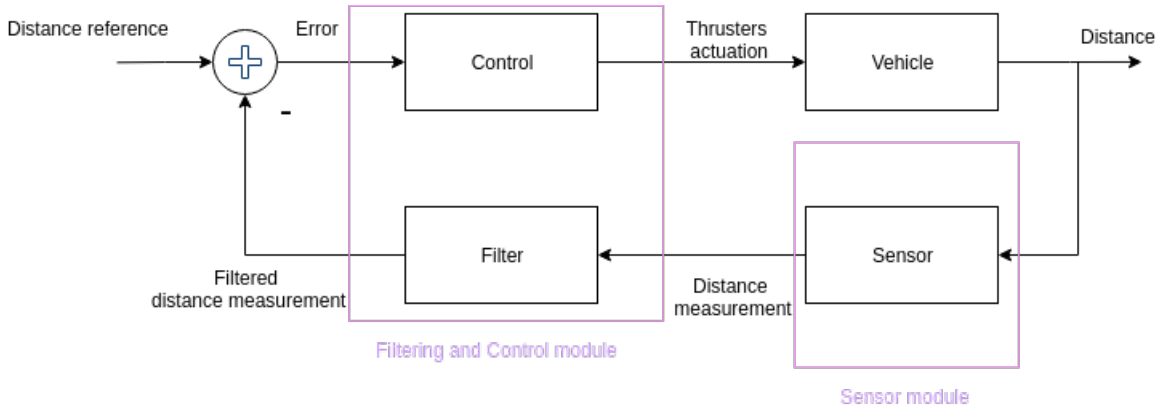

Figure 3.1: Block diagram of the overall system integrating the modules developed in this dissertation

that allows its use in other underwater applications other than the control of the distance of an underwater vehicle.

Unlike the Sensor module, the Filtering and Control module was designed and developed based on the vehicle's properties and the characteristics of the data provided by the Sensor module since its main purpose is the adaptation of the information provided by the Sensor module and its use in order to control the vehicle.

In this dissertation, the underwater vehicle used to test the modules developed was a vertical profiler developed in [1]. The vehicle has approximately 1.35 m of length and a mass of 11.3 kg and its main purposes are the navigation in the vertical axis in order to acquire information related to the water properties and the inspection of the sea floor. An image of the vehicle mentioned is presented in the figure 3.2

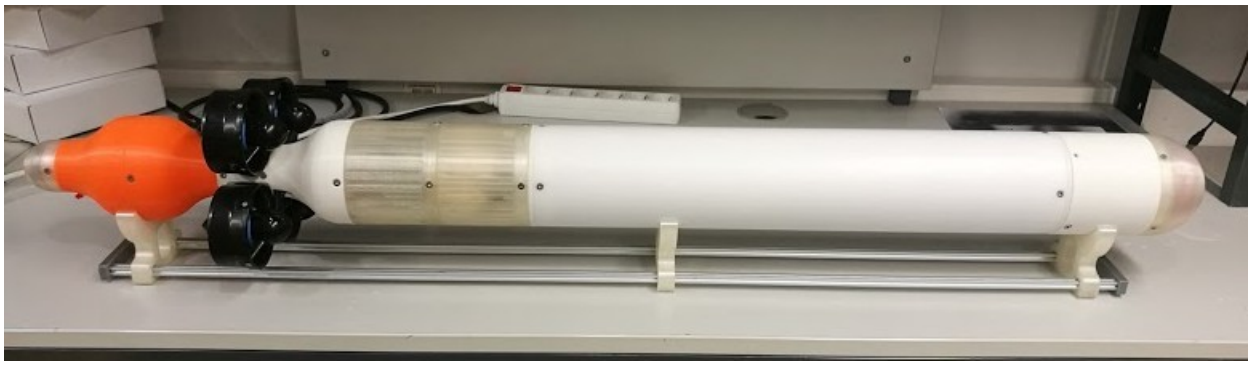

Figure 3.2: Image of the profiler

Based on the area of application and other limitations related to the vehicle and available material, a set of requirements were imposed to the overall system that was distributed by the two modules. These are presented in the following sections. It is also presented the implementation of those modules and the respective tests performed.

3.2 Sensor module

3.2.1 Requirements

The Sensor module's main purpose is the measurement of the distance of the vehicle towards the obstacle based on computer vision. In order to achieve this functionality and guarantee the possibility of utilization of the sensor module in other underwater applications, the module should be able to accomplish the following main requirements:

- Since the profiler's available space is extremely restricted, especially in its extremity where the Sensor module hardware would be placed, some limitations are imposed on its components. Specifically, the hardware needed to implement the Sensor module must have less than 12cm of diameter;
- Must use the already chosen camera for the profiler. In 3.2.2.1 more details of the already integrated camera will be given. Since this camera uses the Vimba SDK (Software development kit), the software responsible for its configuration and frame acquisition must take into account the software development kit of this camera;
- Provide information about the quality of the measurements acquired;
- It must be possible to dictate the information provided by the Sensor module and be possible to control the operations that are performed;
- The software related to the Sensor module must work as an independent process while running and retrieve data via User Datagram Protocol (UDP) communication;
- Considering the applications that this sensor could be used for, it must produce (acquisition and processing) new measurements at a rate of at least 5 measurements per second.

3.2.2 Solution implemented

3.2.2.1 Hardware

In order to achieve the Sensor module functionality required, it is first necessary to define the hardware components. The sensor device was implemented based on two laser pointer devices placed parallel to one another beside a CCD camera. The choice of using laser pointer devices over all other variations of lasers presented in chapter 2 is mainly supported by the inherent lower computer demanding and complexity. Relative to the wavelength of the lasers, the software implemented to detect the laser dots allow the utilization of the green coloration. That range of wavelength was chosen since it has the best overall performance in underwater laser application as shown in chapter 2.

The camera used was the already integrated camera on the profiler, which is a Mako G-125 camera based on the Vimba SDK from Allied Vision [39]. It is well compacted, allows the powering over Ethernet, and in full resolution is able to provide 30.3 frames per second. Although it is

able to provide the mentioned frame rate, an underwater Ethernet cable was used in order to gather the information limiting the maximum frame rate to approximately 9 frames per second.

Relative to the laser pointer devices, it was used two green (532nm) laser modules from Odic-Force Lasers [40]. It was performed an adjustment in order to by-pass the switch pressure button in order to work in continuous operation whenever power is applied.

3.2.2.2 Distance calculation method

The solution implemented to calculate the distance of the vehicle towards the obstacle was based on the laser triangulation principle. This method is illustrated in figure 3.3.

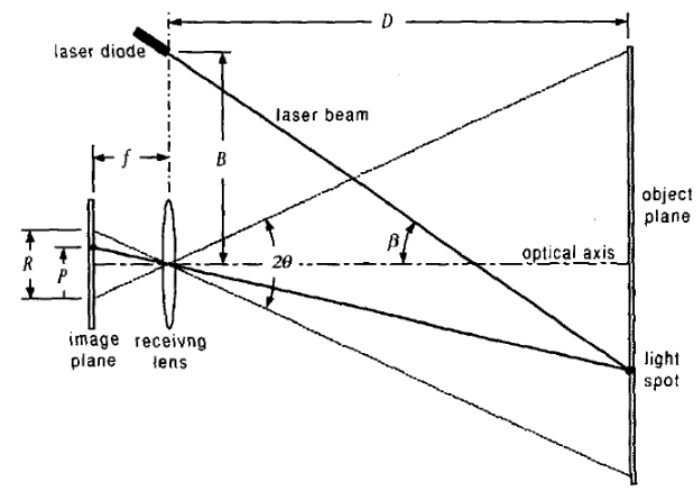

Figure 3.3: Triangulation method using a laser beam and a video camera [4]

.

In figure 3.3, R represents the horizontal resolution in pixels of the camera, f is the horizontal focal length measured in pixels, β the angle between the laser beam and the optical axis, P the pixel where the laser point is projected on the image plane and B the fixed distance between the optical axis and the laser pointer.

The distance to the obstacle (D) can then be given by:

$$\tan\Theta = \frac{R}{2f} \tag{3.1}$$

$$\frac{2P-R}{2f} = \frac{D\tan\beta - B}{D} \tag{3.2}$$

$$D = \frac{BR}{R\tan\beta - (2P-R)\tan\Theta} \tag{3.3}$$

Analyzing the technology involved, the calculation using only one laser pointer device is enough to retrieve the pretended distance. The choice of using two laser pointer devices in the

sensor device is justified by the introduction of redundancy, increasing the reliability of the system and the quality of the measurements by merging the 2 values.

3.2.3 Software architecture

In order to accomplish the requirements presented for the Sensor module, the software developed was structured to perform 4 main features: Acquisition of the frame; Calculation of the distances; Data handling; Communication.

The first feature is responsible for the configuration and communication with the camera device and the acquisition of new frames. Each time a new frame is ready to be used, it is sent to be processed.

The Calculation of the distances feature purpose is the process of the image captured in order to acquire the distance towards the object. It is divided into 3 phases: detection of the laser dots on the image; calculation of the centroids of the laser dots; triangulation method in order to compute the distance. First, the image is processed using image processing methods in order to obtain a left and right segmented image containing only the left and right detected laser respectively. Then, the centroid of each laser dot is obtained and sent to the last phase where based on laser triangulation, a distance towards the obstacle is obtained for each dot in use.

The Data handling is responsible for two main operations: calculation of the quality of the detection of the laser dots and application of certain operations based on the software configuration, for example, application of a circular average on the measurements.

All the information acquired is then sent to the user through UDP communication in the last phase and the software awaits a new frame to process (Communication).

Figure 3.4 illustrates the procedures that occur in a generic execution of the software from the acquisition of the frame to the dispatch of the data to the user.

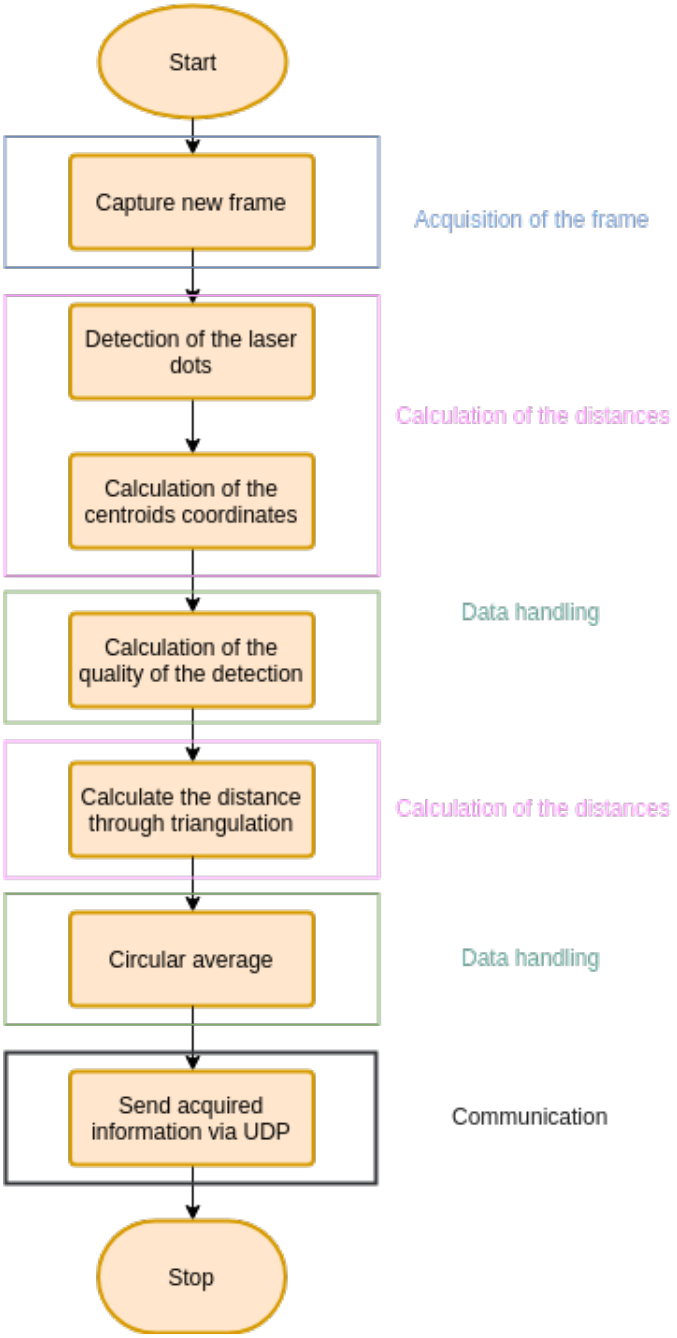

Figure 3.4: Example of a generic high-level execution of the software related to the Sensor module

.

3.2.4 Acquisition of the frame

For the implementation of this feature the software presented in [41] was used. It allows the configuration of the parameters of the camera, for example, frames per second, resolution, and exposure time. It also allows the acquisition of the images of any camera that uses the Vimba SDK. Its use in this dissertation was authorized by its authors, but since the software was not developed during this dissertation, its design and implementation is not going to be detailed.

3.2.5 Calculation of distances

The software responsible for the distance calculation is divided into 3 phases: Laser dots detection, centroid calculation, and triangulation calculation. All these 3 phases were implemented in a class called ImageProcessing. Furthermore, this class posses other methods that improve the quality of the operation and increase the amount of control and information that is extracted.

The main methods included in this class are:

- slot confImageProcessing - slot that is called when a signal symbolizing that a UDP frame referent to the configuration of the image processing process was received. This UDP frame is used to enable or disable the service responsible for the distance calculation by the user during the operation;
- loadImageProcessingConfig - method that reads a XML (Extensible Markup Language) file that can be edited in order to configure all the operations and information that are relevant to be acquired;
- slot ProcessImage - slot that is called when a signal symbolizing that a new frame is ready is sent from the acquisition phase;
- calcDistance - method called by the slot ProcessImage that receives the new image in order to initiate the sequence of processing. It is responsible for the calculation of the distances and then for calling the method capable of transmitting the information acquired to the user;
- detectDots - method responsible for the creation of two segmented images that only contains the left and right laser dots detected separately;
- calcCentroidsDots - method that calculates the centroid coordinates of each of the laser dots detected;
- Triangulation - method that applies the triangulation operation in order to compute the distances;
- calcQualityFactor - method responsible for calculating the quality factor of each measurement;
- calcCircularAverage - method that calculates the circular average of the last N measurements. The value of N is specified through the configuration file.

3.2.5.1 Laser dots detection

The method detectDots is responsible for the laser dots detection. The order of operations is as described in figure 3.5:

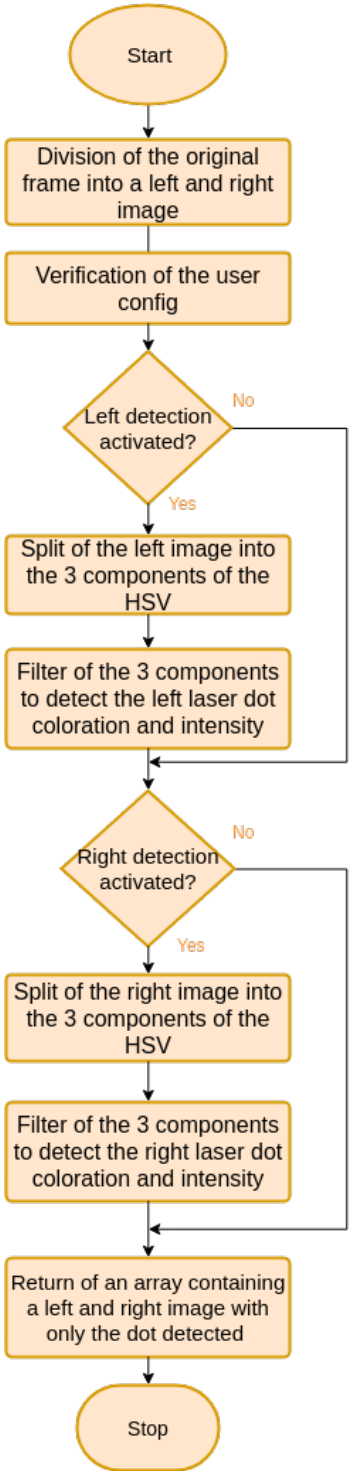

Figure 3.5: Flowchart of the operations that occur in the detectDots method

.

3.2.5.2 Centroid calculation

The method calcCentroidsDots is responsible for the calculation of the centroid coordinates. The order of operation is present in figure 3.6:

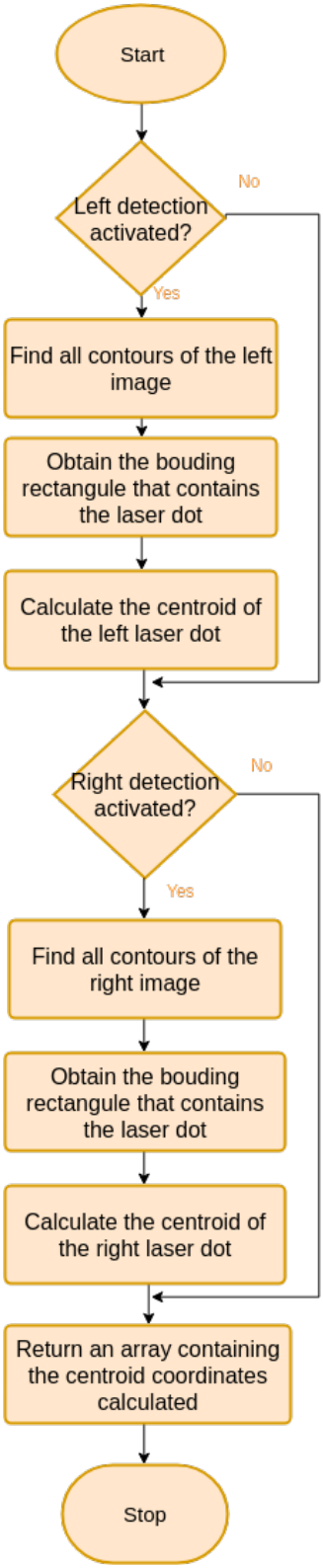

Figure 3.6: Flowchart of the operations that occur in the calcCentroidsDots method

.

3.2.5.3 Triangulation calculation

The computation of the distances is executed by the Triangulation method. Figure 3.7 shows the sequence of actions that take place in this calculation.

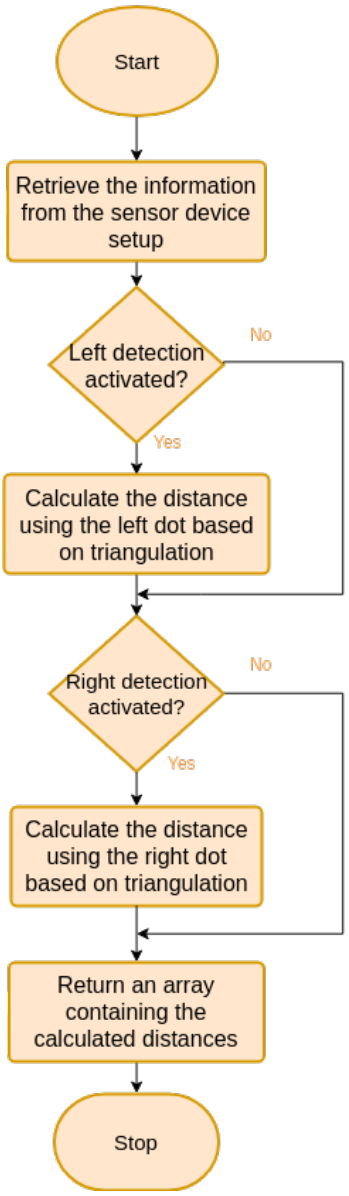

Figure 3.7: Flowchart of the operations that occur in the Triangulation method

.

3.2.6 Data handling

After the computation of the centroid coordinates of the laser dots, the method calcDistance calls the method calcQualityFactor that retrieves a quality factor of each laser dot detected based on its location and circularity. If the quality factor is considered lower than a determined threshold, the detection of that specific laser dot is treated as not successful and the distance related to that laser

dot is not calculated. After the calculation of the distances by the Triangulation method, the results of the detection and the distances calculation are examined in order to decide what information to send to the user. Each operation has the capacity of detection errors and inserting error codes in the procedure that can be detected by this phase. The information sent is dependent on how successfully the calculation was.

3.2.7 Communication

After the determination of the information that is going to be sent to the user, an array is filled with all the information demanded in the software's configuration. The format of this array is as follows:

Table 3.1: Communication array format

Vector position	Information
0	Left dot distance
1	Right dot distance
2	Average or circular average
3	Quality factor of the left dot detection
4	Quality factor of the right dot detection

The information of this array depends on the configuration of the software, quality of the measures and on how successfully the overall procedure was. For example, if the left or right detection is disable on the configuration, in the positions 0 or 1, correspondingly, the user will receive the value -2, indicating that the functionality was not enabled in the configuration file. The same is verified for the rest of the array positions since the average, circular average and quality factors calculations can be enabled or disabled in the configuration file.

In case of any type of error detected during the calculation of any of the information of those positions, the user will receive the value -1, indicating that the data acquired for that array position was considered not reliable nor valuable. This can occur in a situation of internal errors, fail on the detection of the laser dots or centroid coordinates or if the quality factor is considered below a certain value.

After filling the array mentioned, this array is sent via UDP communication. This functionality was implemented by modifying and adding necessary methods to a class present in the software used in [41] already mentioned, class named Protocol. This class originally allowed the features of sending and receiving UDP packages. Minor changes were made to some methods already present in this class and the following 3 methods were added:

- A method that allows the user to initiate and terminate the calculation of the distance through UDP communication (named send_ConfigImageProcessing);

- A method used by the calcDistance method in order to fill a package that contains the array of information following certain rules to make it clear for the user the order of the information transmitted (called send_DistCalc);

- A method that allows the user to receive the information calculated by the Sensor module through UDP (ReceiveDistance).

The class Protocol mentioned and a XML file are the interface elements that allow a user to communicate and control the Sensor module. It is needed of the method socket present in the Protocol class in order to create a socket capable of communicating with the Sensor module and the use of the method send_ConfigImageProcessing containing the command to enable or disable the calculation of the distance. A C++ example of this procedure is illustrated in figure 3.8.

```
/// Send the signal to config the ImageProcessing
std::string IP("127.0.0.1");
Protocol socket( IP, 4665, 4666);/*1st arg -> My ip ; 2nd arg -> port_rcv ; 3rd arg -> port_send */

int config = 1;
switch(config){

case 0:
    std::cout << "Config value: 0, only camera activation signal sent" << std::endl;
    break;
case 1:
    std::cout << "Config value: 1, camera and distance calculation activation signal sent" << std::endl;
    socket.send_ConfigImageProcessing(1);
    break;
}
```

Figure 3.8: Example of the interaction with the Sensor module in order to enable its operation

.

Furthermore, an XML file present in the Sensor module package must be filled as shown in figure 3.9.

```
<?xml version="1.0"?>
<opencv_storage>
        <ResolutionWidth>1292</ResolutionWidth>
        <ResolutionLength>964</ResolutionLength>
        <focal_mm>6</focal_mm>
        <sensorWidth_mm>4.8</sensorWidth_mm>
        <angle>0</angle>
        <leftLaser>1</leftLaser>
        <rightLaser>1</rightLaser>
        <enableAverage>0</enableAverage>
        <recieveBothMeasures>1</recieveBothMeasures>
        <nSamplesToAverage>5</nSamplesToAverage>
        <recieveQualityFactor>1</recieveQualityFactor>
</opencv_storage>
```

Figure 3.9: Configuration XML file example

.

These steps will initiate the Sensor module operation. In order to receive the information, the user uses the ReceiveDistance method present in the Protocol class as presented in the C++ example in figure 3.10.

```
double infoReceived[5];
infoReceived[0] = -1; // left
infoReceived[1] = -1; // right
infoReceived[2] = -1; //Average or CircularAverage
infoReceived[3] = -1; //LeftQF;
infoReceived[4] = -1; //RightQf

socket.ReceiveDistance(infoReceived);
```

Figure 3.10: Example of the use of the ReceiveDistance method

.

3.2.8 Characterization and tests

In order to characterize and test the sensor module, it is first necessary to know the limitation of the technology used in the distance calculation. Through the triangulation method it is known that the accuracy of the measurements is inversely proportional to the distance of the obstacle. This can be demonstrated by the analysis of the distance procedure shown in 3.2.2.2:

$$D = \frac{BR}{R\tan\beta - (2P - R)\tan\Theta} \tag{3.4}$$

Considering the angle between the laser beam and the optical axis (β) equal to 0, the equation presented in 3.5 is given by:

$$D = \frac{Bf}{\frac{R}{2} - P} \tag{3.5}$$

The denominator on the equation presented represents the distance in pixels between the pixel where the laser point is projected on the image plane and the center point of the image ($\Delta\, d_{pixels}$).

Since the value of B and f are constant values related to the setup of the sensor device, the only determinant factor is the value of $\Delta\, d_{pixels}$. Since this value depends only on which pixel the laser dot is detected, it is then possible to assert that the range of values of the distance obtained is "discrete" and has an error equal to the distance necessary to go through that makes the detection of the laser to be in the forthcoming pixel. In table 3.2 it is illustrated the value of the inherent error caused by the use of this method for 4 different distances for B equal to 4cm and f equal to 1615. The values for B and f have no practical meaning and are therefore only valuable in order to exemplify the topic described.

Using the results from table 3.2 is possible to verify that the error is increasing with the increase on the distance as expected.

It is also necessary to know the error induced through the utilization of non-ideal components and possible software limitations. In order to quantify these errors, a methodology of testing is

Table 3.2: Inherent error caused by laser triangulation

Δ d_{pixels}	D for Δ d_{pixels}	Δ d_{pixels} + 1	D for Δ d_{pixels} + 1	error	error(%)
250	0.258m	249	0.259m	0.1cm	0.39
60	1.077m	59	1.095m	1.8cm	1.69
36	1.794m	35	1.846m	5.2cm	2.90
21	3.076m	20	3.230m	15.4cm	5.00

used to ensure that it is possible to not only calibrate the sensor module but also examine the possible error present on the measures.

The structure set up in order to test the Sensor module is illustrated in the images 3.11 and 3.12.

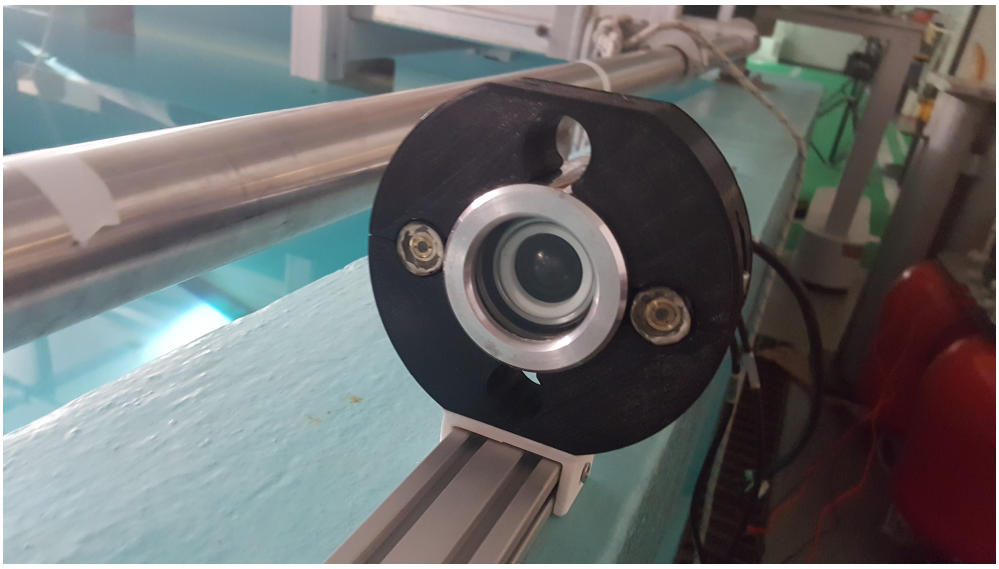

Figure 3.11: Structure containing the camera and the laser devices

.

The structure illustrated contains the camera, laser pointer devices mentioned in 3.2.2.1 fixed into a 3 *m* structure marked each 10 *cm* and a movable target.

The first phase of tests is the determination of the divergence of the camera relative to the target. In order to do so, the target contains a center mark that can be compared to the center of the image that is being captured by the camera. A series of tests were performed from the range of 10 to 250 cm, moving the target 20cm between each test. The results obtained are presented in table 3.3.

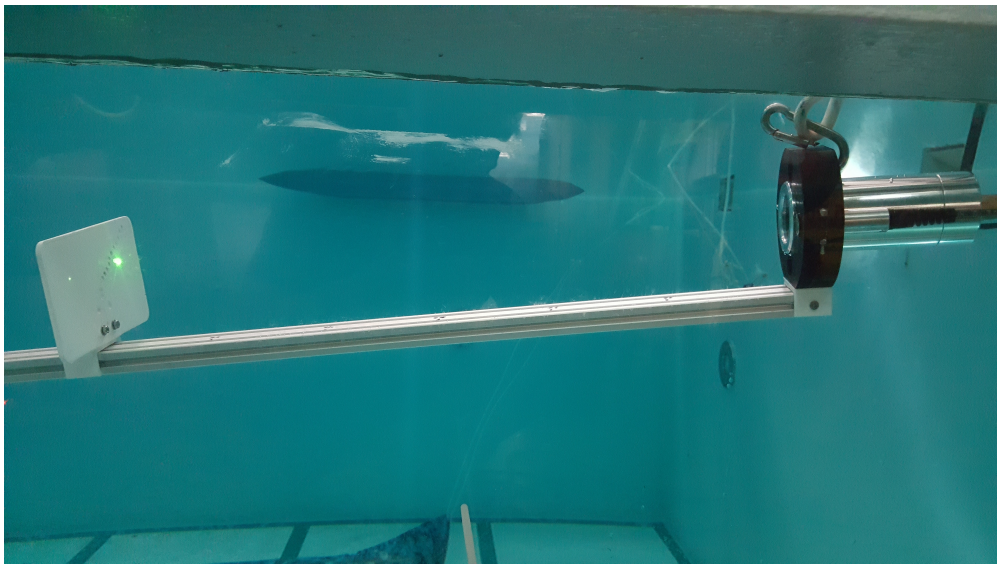

Figure 3.12: The full structure underwater, containing the target on the left
.

Table 3.3: Deviation of the camera angle of view

D	Deviation of the center
0.10 m	0.3 cm
0.30 m	0.7 cm
0.50 m	1.1 cm
0.70 m	1.6 cm
0.90 m	2.1 cm
1.10 m	2.6 cm
1.30 m	3 cm
1.50 m	3.4 cm
1.70 m	3.7 cm
1.9 m	4.1 cm
2.1 m	4.4 cm
2.3 m	4.8 cm
2.5 m	5.1 cm

By analyzing the information from table 3.3 it is possible to calculate a linear regression that translates the increment of the deviation with the increase of the distance towards the obstacle. The equation calculated capable of defining the deviation in meters is presented in 3.6.

$$Deviation = 0.02033 \times D + 0.001956 \tag{3.6}$$

The second step is the calculation of the divergence of each laser pointer device. The laser

devices were placed with 4cm to the center each and it is expected that as the distance increases this distance on the target will change due to the angle of the laser devices. The measurement of this divergence took into account the deviation of the camera. For each test, the distance of the laser beams on the target towards the center ($D_{\text{left-center}}$ and $D_{\text{right-center}}$) was calculated and the results are presented in table 3.4.

Table 3.4: Deviation of the lasers

D	$D_{\text{left-center}}$	$D_{\text{right-center}}$
0.10 m	4.2 cm	3.8 cm
0.30 m	4.4 cm	3.5 cm
0.50 m	4.7 cm	3.2 cm
0.70 m	5.0 cm	2.9 cm
0.90 m	5.3 cm	2.6 cm
1.10 m	5.5 cm	2.4 cm
1.30 m	5.7 cm	2.2 cm
1.50 m	6.0 cm	1.9 cm
1.70 m	6.3 cm	1.7 cm
1.9 m	6.7 cm	1.4 cm
2.1 m	7.0 cm	1.1 cm
2.3 m	7.3 cm	0.8 cm
2.5 m	7.5 cm	0.6 cm

As expected the rate of change of the distance of the lasers towards the center is constant due to the angle of the lasers being constant. Therefore, it is possible to calculate a linear regression of this divergence and use it in order to correct it. The linear regressions obtained were the following:

$$D_{\text{left-center}} = 0.01404 \times D + 0.03990 \tag{3.7}$$

$$D_{\text{right-center}} = -0.01316 \times D + 0.03872 \tag{3.8}$$

After these two procedures of calibration, a series of tests were performed from the range of 10 to 250 cm, moving the target 20cm between each test in order to validate the performance of the Sensor module. The deviation study performed previously was used in order to correct the distance of the lasers towards the center at each calculation of the distance. This means that initially, the software uses the value of 4 *cm* for the distance of the lasers, but after each distance calculation, the value obtained is inserted in the linear regression equations in order to retrieve the value of the actual deviation of the laser dots towards the center.

The results obtained for the distance calculated using the left laser dot (D_l) and right laser dot (D_r) are presented in the table 3.5.

Table 3.5: Results obtained for sensor performance

D	D_l	D_r	$Error_l$	$Error_r$
0.10 m	0.1004 m	0.1014 m	0.4 mm	1.4 mm
0.30 m	0.3011 m	0.3017 m	1.1 mm	1.7 mm
0.50 m	0.4942 m	0.5027 m	5.8 mm	2.7 mm
0.70 m	0.7022 m	0.6889 m	2.2 mm	1.1 cm
0.90 m	0.9106 m	0.8769 m	1.06 cm	2.31 cm
1.10 m	1.0832 m	1.0737 m	1.68 cm	2.63 cm
1.30 m	1.2795 m	1.2785 m	2.05 cm	2.15 cm
1.50 m	1.4908 m	1.4712 m	0.92 cm	2.88 cm
1.70 m	1.6958 m	1.7196 m	0.42 cm	1.96 cm
1.9 m	1.8983 m	1.8742 m	0.17 cm	2.58 cm
2.1 m	2.1740 m	2.1606 m	7.4 cm	6.06 cm
2.3 m	2.3579 m	2.2253 m	5.79 cm	7.47 cm
2.5 m	2.5635 m	2.4225 m	6.35 cm	7.75 cm

The results show that the sensor is working as intended showing an error within the range mentioned in table 3.2, where the deviations compared to that range is essentially due to the infrastructure, where the camera, obstacle and even the lasers might have suffered small changes in their position during the tests. Furthermore, some error in the measurement of the lasers and camera deviations that were acquired manually have a significant impact on the measurements obtained.

In images 3.13 and 3.14 a better visualization of the error obtained in these tests is presented.

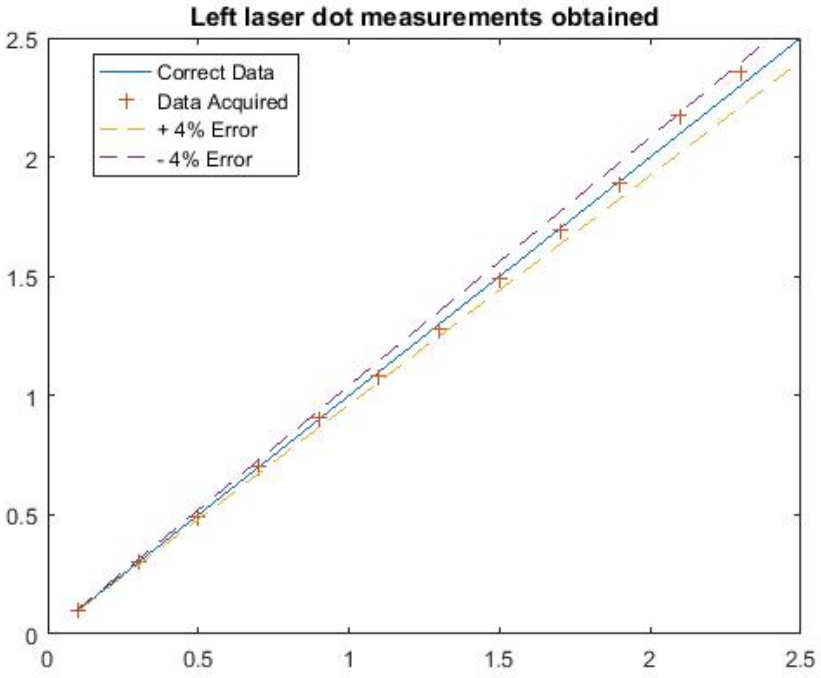

Figure 3.13: Measurements obtained by the left laser dot

.

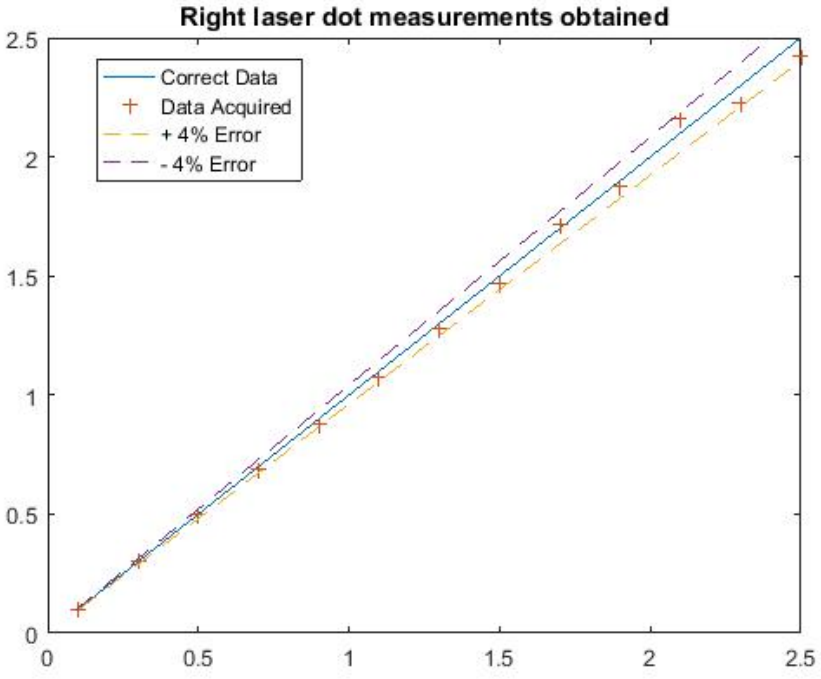

Figure 3.14: Measurements obtained by the right laser dot

.

Through the observation of the graphics presented, it is possible to assert that the measurements obtained presented an error inferior to 4 % for the full range of measurements.

Since 2 measurements were acquired for the same altitude, it is opportune to verify if a simple average operation between the measurements would benefit the overall accuracy of the acquisition.

Table 3.6 showcases the relative error in percentage for each altitude point for each laser point (E_l and E_r), as well as the relative error in percentage after the application of the average operation (E_avg).

Table 3.6: Relative error for each altitude point and for the average between the measurements

D	E_l	E_r	E_{avg}
0.10 m	0.4610	1.4350	0.9480
0.30 m	0.3667	0.5547	0.4607
0.50 m	1.1620	0.5360	0.3130
0.70 m	0.31	1.6046	0.6473
0.90 m	1.1756	2.5624	0.6934
1.10 m	1.5245	2.3945	1.9595
1.30 m	1.5738	1.6507	1.6122
1.50 m	0.6154	1.9207	1.2680
1.70 m	0.25	1.1524	0.4512
1.9 m	0.3511	1.3596	0.8554
2.1 m	3.5256	2.8869	3.2063
2.3 m	2.5173	3.2478	0.3652
2.5 m	2.54	3.0996	0.2798

The average relative error of the left and right measurements was 1.2595 % and 1.8773 %, respectively. The average relative error after the application of the average operation using both measurements was equal to 1 %, showing a considerable improvement.

3.3 Filtering and Control module

3.3.1 Requirements

The Filtering and Control module is responsible for the filtering of the input data given by the Sensor module and posterior control of the vehicle distance towards the object using that information. The following requirements were imposed to this module:

- The filter must be able to reject outlier measurements gathered by the Sensor module;
- The filter must ensure that the control is reliable even when no measurement was acquired by the Sensor module. This means that the state of the system must be updated even when the Sensor module could gather reliable information.

- The controller must be flexible based on the necessities and characteristics of the pretended mission. This means that the parameters of the controller must be able to be easily modified if the mission requires more agressive or smooth actuation, or if there are time limitations or energetic demands;
- The controller must be easily adjusted if the vehicle's properties change.

Since the Filtering and Control module is meant to deal with the filtering of the received data and the control tasks, it was divided into two phases: Interface and Filtering phase, and Control phase.

3.3.2 Interface and Filtering phase

The Interface and Filtering phase is responsible for 3 tasks: communicate with the Sensor module in order to retrieve the information gathered; apply a Kalman filter to the data received; transmit the estimates to the vehicle. This phase serves as the link between the vehicle software control and the Sensor module software.

The first task starts with a creation of a separated process that communicates with the sensor and receives the periodic data. This procedure is done following the steps mentioned in section 3.2.7.

The main role of this phase is the use of a discrete Kalman filter in order to handle the data received, creating estimated information that can be used by the control phase.

The Kalman filter implemented has 3 state variables, predicting and updating the altitude, depth, and velocity. These states are fed by the Sensor module and by the pressure sensor of the profiler.

The filter implementation was based on the conventional equations of a Kalman Filter [42] considering that there is no noise added to the process.

Discrete Kalman filter time update equations:

$$\hat{x}_k^- = A\hat{x}_{k-1} \tag{3.9}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{3.10}$$

Discrete Kalman filter measurement update equations:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{3.11}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(Z_k - H\hat{x}_k^-) \tag{3.12}$$

$$P_k = (I - K_k H)P_k^- \tag{3.13}$$

In the equations presented, $\hat{x}_k^-$ represents the matrix of the prediction values in the current time update iteration, A is the matrix representative of the evolution of the state, $\hat{x}_{k-1}$ is the matrix of estimations from the previous iteration. P_k^- is the matrix of the error in the estimation after the time update, P_{k-1} represent the matrix of the error in the estimation from the previous iteration. Q is a matrix of error induced related to the process, K is a matrix of the Kalman filter gains, H is the observation matrix, Z_k is the vector containing the measurements, ,and R is a matrix that represents the error in the measurements.

Since this dissertation's main objective is the control of the altitude of the profiler when it moves vertically, the filter was designed in a way that allows the update of the altitude of the vehicle even if the Sensor module could not retrieve any measurements. This is possible since the variation of the depth is the inverse of the variation of the altitude when the profiler moves vertically and therefore in cases where the sensor does not calculate any reliable measurements, the variation of the depth is used in order to have an approximate update of the altitude value.

Furthermore, since this filter is going to transmit the estimations to the control of the vehicle, there is a special attention given to the rate of smoothing that the Kalman filter applies. Typically, the Kalman filter is used in order to have the best estimation possible of the state variables, but in this application, the matrix Q is tuned in order to allow a higher or lower variation of the measurements in order to reduce the abrupt variation on the measurement on the input of the controller. The values in the matrix Q depend not only on the error of the process but also in the error of the sensors that are used.

The values of the matrix A, P, and H are constant and are chosen based on the process and knowledge of the error in the initial state:

$$A = \begin{bmatrix} 1 & 0 & -T \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The values of the matrix A show that it is taken into account the velocity in z direction using the pressure sensor when predicting the value of the next altitude since it is considered that the profiler moves mainly in the vertical axis. The variable T represents the period of the Kalman filter

execution. The values of the matrix P are meant to introduce a significant error at the start since it is not known the initial condition. The values of the initial state ($x0$), Q and R are dependent on the mission and the sensors that are used.

Lastly, this phase is responsible for the communication with the already existing profiler's software in order to update the distance value of the vehicle. This is done through a shared memory block already implemented in the profiler's software. In [1] more information is given relative to the implementation of the shared memory block. A new position on the shared memory block was added and its value is updated every time the Kalman filter is executed and calculates a new estimate.

3.3.3 Control phase

This phase is responsible for the determination of the actuation necessary to impose on the thrusters in order to achieve the reference of altitude on the input of the control loop.

The solution adopted regarding the control is based on the switching of two controllers, a velocity controller, and a position controller. The idea is that the vehicle moves with a specified velocity imposed using the velocity reference while it is still considered far away from the reference and then a position controller is used to complete the movement and reach the position reference. This allows more control over the movement of the vehicle through the mission especially in the velocity of the movement and avoids the application of high value steps into a position controller that would result in a high power consumption. Additionally, this division allows the design of the velocity and position controllers based on the demands that each phase of the movement requires, approximation and reach of the reference, respectively.

In order to design and validate the controllers through simulation, the approximate mathematical model of the dynamics of the vehicle related to its movement must be calculated.

3.3.3.1 Vehicle model

In [1] the model of the vehicle for its 6 degrees of freedom was obtained following the procedures presented in [5]. The results obtained in [1] related to the vehicle's model and its experimental parameters were used in this dissertation in order to retrieve the equation that expresses its movement along the body fixed z- axis since that its control is the focus of this dissertation.

In figure 3.15 it is illustrated the reference frames that were considered.

In table 3.7 is presented the notation used in [5] for marine vehicles.

The overall idea of the procedures presented in [5] is that the motion of an underwater vehicle can be described in the body fixed frame using the equation 3.14:

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau \tag{3.14}$$

where M is the inertia matrix including added mass, $C(v)$ represents the matrix of Coriolis and centripetal terms including added mass, $D(v)$ is the damping matrix, $g(\eta)$ is the vector of

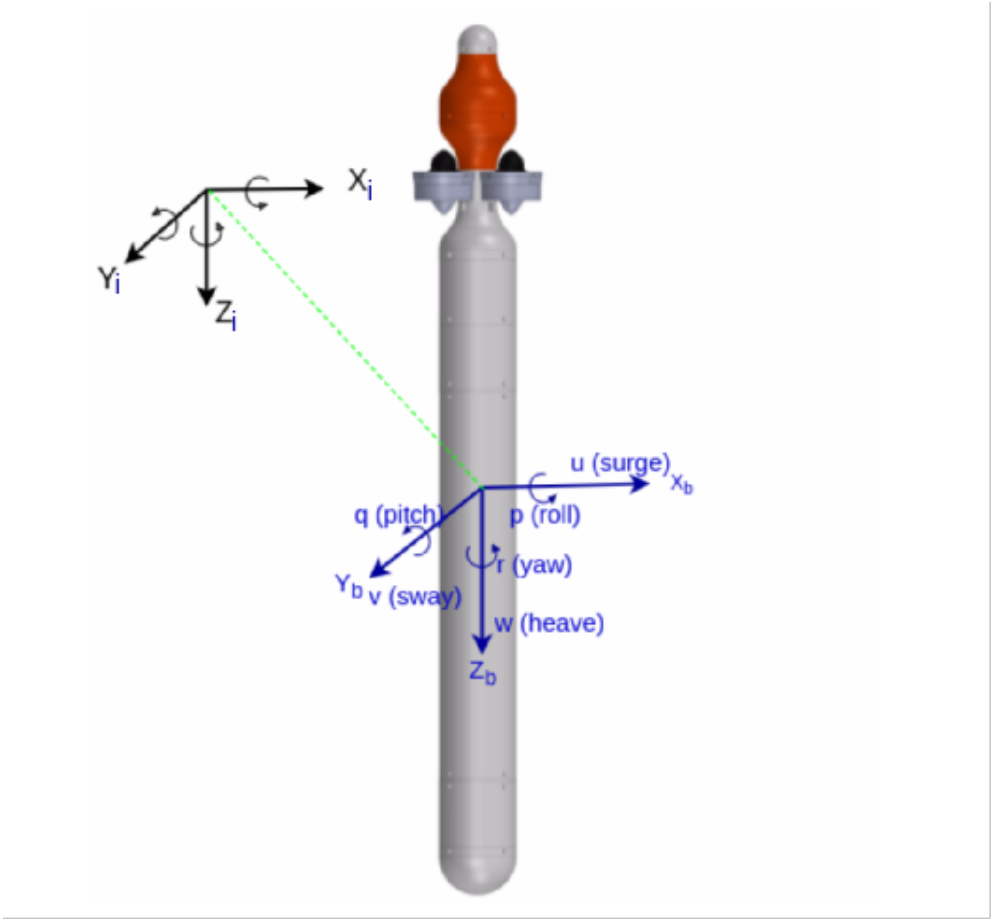

Figure 3.15: Vehicle's reference frames considered [1]

gravitational forces and moments and τ is the vector of control inputs. In [1] more detailed information about the deduction of the vehicle's model is available.

Extracting the equation relative to the variation of motion in the z-direction ($\dot{w}$):

$$\dot{w} = 0.086(Z + Z_{ww}w^2 + Z_w \times w - F_{flotation}) \tag{3.15}$$

where Z_w and Z_{ww} are the linear and quadratic damping coefficients, and $F_{flotation}$ is the flotation force imposed in the vehicle. The value of these coefficients were obtained from the experiment results done in [1].

Table 3.8: Necessary parameters for the vehicle's mathematical model

Parameter	Value	Units
Z_{ww}	-0.92	kg/m
Z_w	-0.92	kg$\times m/s$
$F_{flotation}$	-1.176	N

Table 3.7: Notation used for marine vehicles [5]

DOF		forces and moments	linear and angular vel.	positions and Euler angles
1	motion in the x-direction (surge)	X	u	x
2	motion in the y-direction (sway)	Y	v	y
3	motion in the z-direction (heave)	Z	w	z
4	rotation about the x-axis (roll)	K	p	ϕ
5	rotation about the y-axis (pitch)	M	q	θ
6	rotation about the z-axis (yaw)	N	r	φ

The next step is the linearization and calculation of the state space of the model presented. The linearization is necessary since the model has a quadratic factor related to the damping force.

In this specific application it was determined that the values of the possible step values that would be imposed to the velocity controller are in the range from 0.05 to 0.2 *m/s*. This is justified and explained in 3.3.3.2.

Based on this assumption, the point of equilibrium chosen was equal to 0.15 *m/s*. After the calculation of the linearized model using this point, the model was also linearized around other 3 points of equilibrium in order to evaluate the impact of the choice of the equilibrium point. The results are presented in table 3.9 and the analysis of those values shows that the difference in the model obtained is negligible for the different equilibrium points.

The following procedures illustrate the steps performed to obtain the state space:

$$x_1 = d \tag{3.16}$$

$$x_2 = w \tag{3.17}$$

$$f_1 = \dot{x_1} = w \tag{3.18}$$

$$f_2 = \dot{x_2} = \dot{w} \tag{3.19}$$

$$y = d \tag{3.20}$$

$$u = Z \tag{3.21}$$

$$\dot{x} = Ax + Bu \tag{3.22}$$

$$\dot{y} = Cy \tag{3.23}$$

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{f_1}}{\partial x_1} & \frac{\partial \dot{f_1}}{\partial x_2} \\ \frac{\partial \dot{f_2}}{\partial x_1} & \frac{\partial \dot{f_2}}{\partial x_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{\partial \dot{f_1}}{\partial u} \\ \frac{\partial \dot{f_2}}{\partial u} \end{bmatrix} \begin{bmatrix} u \end{bmatrix} \tag{3.24}$$

$$\begin{bmatrix} \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{y}}{\partial x_1} & \frac{\partial \dot{y}}{\partial x_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.25}$$

The result of the state of space for the linearized model around w* = 0.15 is given by:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -0.1029 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.0860 \end{bmatrix} \begin{bmatrix} u \end{bmatrix} \tag{3.26}$$

$$\begin{bmatrix} \dot{y} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{3.27}$$

Based on the state of space the transfer function that represents the behavior of the vehicle is calculated:

$$G(s) = C(sI - A)^{-1}B \tag{3.28}$$

Substituting the values, the following transfer function is obtained:

$$G(s) = \frac{0.0860}{s(s + 0.1029)} \tag{3.29}$$

The transfer function obtained represents the relation between the input given to the thrusters of the vehicle and the variation of z. Based on this transfer function it is easy to conclude that a transfer function that relates the input given to the thrusters of the vehicle and the variation of w can be given by:

$$G(s) = \frac{0.0860}{s + 0.1029} \tag{3.30}$$

Table 3.9: Variation of the coefficients of the linearized models using different points of equilibrium

Point of linearization	Gain	Pole
0.2 m/s	0.0860	0.1108
0.15 m/s	0.0860	0.1029
0.10 m/s	0.0860	0.0950
0.05 m/s	0.0860	0.0871

3.3.3.2 Implementation of the position controller

The method created to tune the controllers starts with the design of the position controller. In this phase, the two main objectives are the design of a position controller that guarantees no overshoot and a good settling time considering the slow dynamics of the underwater vehicle movement, and the determination of the point where the controllers switch.

Since the position control is meant to control the altitude of the vehicle it is essential that when the switching point occurs, the Sensor module can calculate reliable information related to the altitude. Therefore, the switching point must occur when the altitude of the vehicle is inferior to the maximum distance that the sensor module can work. Furthermore, the switching point must guarantee that the switching of the controllers is done smoothly avoiding abrupt variations of the actuation provided by the thrusters.

The position controller was implemented based on the conventional PD (Proportional – Derivative) approach. The reason behind this choice is given by the analysis of the transfer function obtained referent to the position of the vehicle. Since it is a second-degree transfer function with a pole in the origin, it is possible to use a PD controller in order to introduce a zero that cancels the plant's pole located in the left plane of the root-locus. The root-locus obtained is given by:

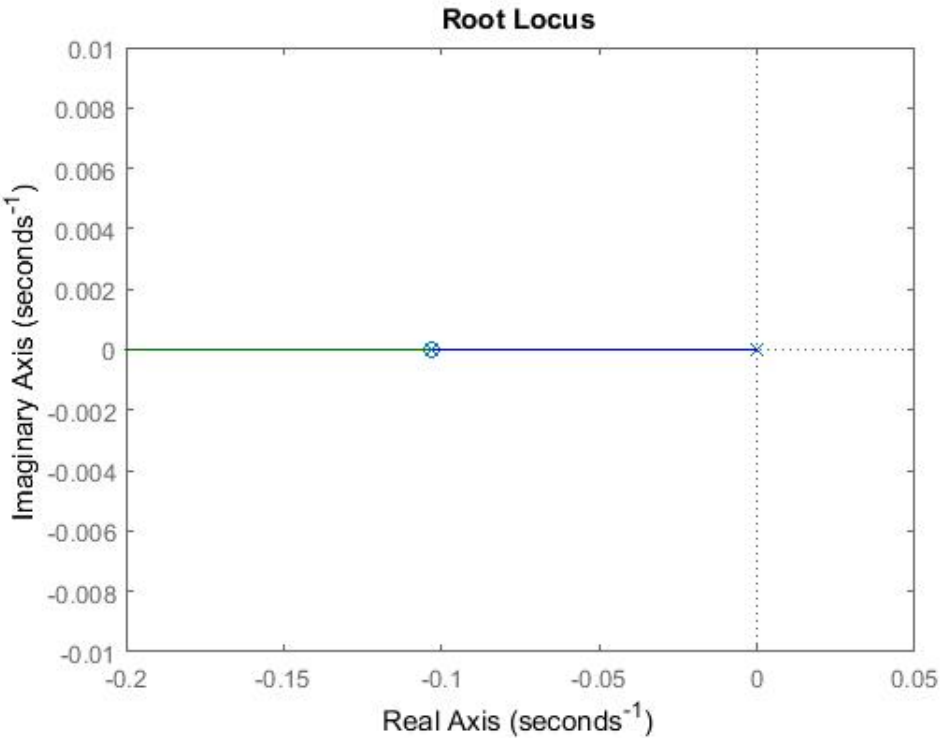

Figure 3.16: Position control root-locus

.

Analyzing the root-locus presented, it is possible to assert that the response to the step of the closed-loop system has no oscillation nor overshoot since the closed-loop pole is always real.

Therefore, this allows the tune of the settling time based on the value of the proportional gain.

The PD controller law obtained is given by:

$$F(s) = K_p \frac{s + 0.1029}{s} E(s) \tag{3.31}$$

The value of K_p was chosen in order to have a settling time below 10s since it was considered that this would be a good value for the performance of the controller considering its transient response and the power required. The step response obtain for K_p equal to 5 is illustrated in figure 3.17.

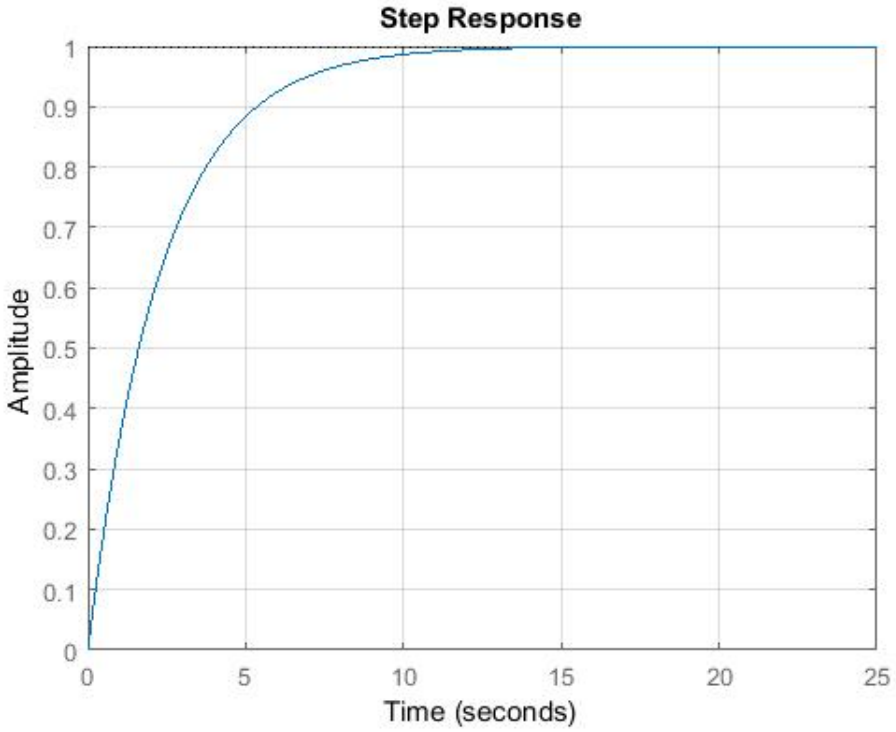

Figure 3.17: Position control response for K_p equal to 5

.

The idea behind the calculation of the switching point is that the actuation that the position controller requires on the switching point is approximately equal to the actuation required by the velocity controller on that exact moment, making the transition smooth for the thrusters.

For this calculation, it was considered that the actuation required by the velocity controller on the switching point is the actuation required in steady-state to maintain the velocity reference, which is sufficient to cancel the flotation and damping force applied on the vehicle. This assumption is justified by the fact that it is possible to assert that when the switch happens the velocity has reached the steady-state since the idea of the velocity controller is to approximate the vehicle when it is far away from the obstacle.

Using the non-linear model calculated in 3.3.3.1 it is possible to determine the actuation required to negate the effect of the flotation and damping forces for a certain velocity through the equation 3.15. Therefore, the value of the actuation that the position controller must actuate on the switching point is calculated for a certain velocity reference.

Since it is considered that the velocity reaches the steady-state before the switching point, the typical control law for the PD:

$$F(t) = K_p e(t) + K_d \dot{e}(t) \tag{3.32}$$

Can now be given by:

$$F(t) = K_p e(t) - K_d V_{ref} \tag{3.33}$$

This is possible because it is known that the rate at which the error is changing is equal to the velocity reference since it is considered that the velocity reached its steady state. The negative sign is justified by the decrease of the error when the velocity is positive.

Since the value of K_p and K_d are known through the design of the position control, for a given velocity response it is possible to know the actuation necessary on the switching point using the equation in 3.15 and therefore know the value of the error that must exist in order to make the position controller require the wanted actuation as shown in equation 3.34.

$$Error = \frac{F_{required} - K_d V_{ref}}{K_p} \tag{3.34}$$

Adding the value of the position reference to the obtained Error value, the switching point is obtained. As stated previously, the value of the switching point must be lower than the maximum distance that the sensor module is capable of acquiring the altitude. Because of this it is necessary to limit the velocity at which the vehicle can be navigating so that the sum of the value of the position reference and the Error value is inferior to the maximum altitude value where the Sensor module is capable of acquiring the distance.

Analyzing the data collected presented in the tables 3.5 and 3.2 it was decided that in this control application, the range where the sensor was going to be used was from 0.10 to 2.5 m since after that value the absolute error was considered not appropriate for this application. Therefore, the velocities accepted are in the range from 0.05 to 0.2 *m/s*, where the value of 0.2 *m/s* would require a switching point of 2.3730 m that is near the maximum altitude where the sensor will be used. This range of possible velocities references is also limited by the position reference input since switching point is the value of the position reference plus the error obtained for the switching point. For example, the choice of the maximum possible velocity (0.2 *m/s*) would require a position reference inferior than 0.127 m.

Since the use of the controller designed in a real application requires its implementation in discrete time, it is necessary to evaluate the response of the system when the controller is in discrete time.

As stated and shown in [43] by selecting a sampling frequency of at least 10 times higher than the system bandwidth it is possible to have a really good approximation of the continuous response with a discrete time controller. In the case of this dissertation the sampling frequency is limited by the value of frames per second that the camera can acquire, which is approximately 9 frames per second, the equivalent of a sampling period of 0.111(1) seconds. The value of 0.1027 rad/s was obtained for the system bandwidth frequency which guarantees that the frequency of acquisition of the camera is greater than 10 times the system bandwidth, and therefore an approximation of the controller in continuous time can be made to the discrete time without affecting significantly its performance.

As shown in [43] the best performance approximation method is the Tustin approximation also called bilinear transformation. This method is based on a trapezoidal approach and can be implemented by substituting the value of s as shown in equation 3.35.

$$s = \frac{2(z-1)}{T(z+1)} \tag{3.35}$$

where T represents the sampling period, which in this case is equal to 0.111(1) seconds.

The result of this approximation and its comparison with the continuous controller response is given in figure 3.18.

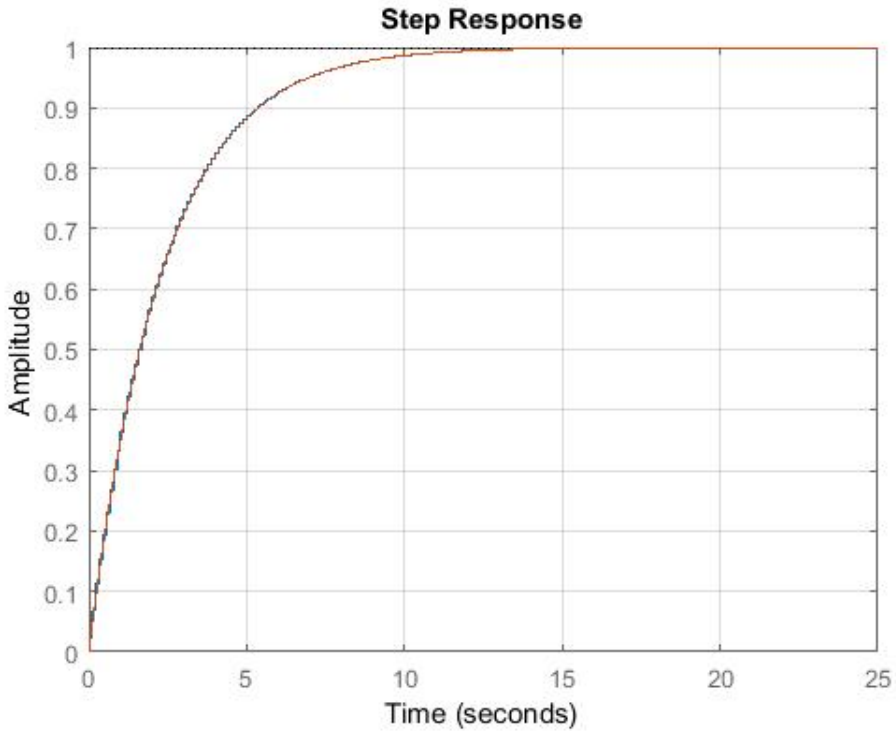

Figure 3.18: Comparison between the response of the system using the continuous and the discrete controllers for position

3.3.3.3 Implementation of the velocity controller

The velocity controller was implemented based on the conventional PI (Proportional-Integral) approach. In the same way of the position controller, this choice is given by the analysis of the transfer function obtained referent to the velocity of the vehicle. Since it is a first-degree transfer function, it is possible to use a PI controller in order to introduce a zero that cancels the plant's pole and add a pole placed at the origin. The root-locus obtained is given by:

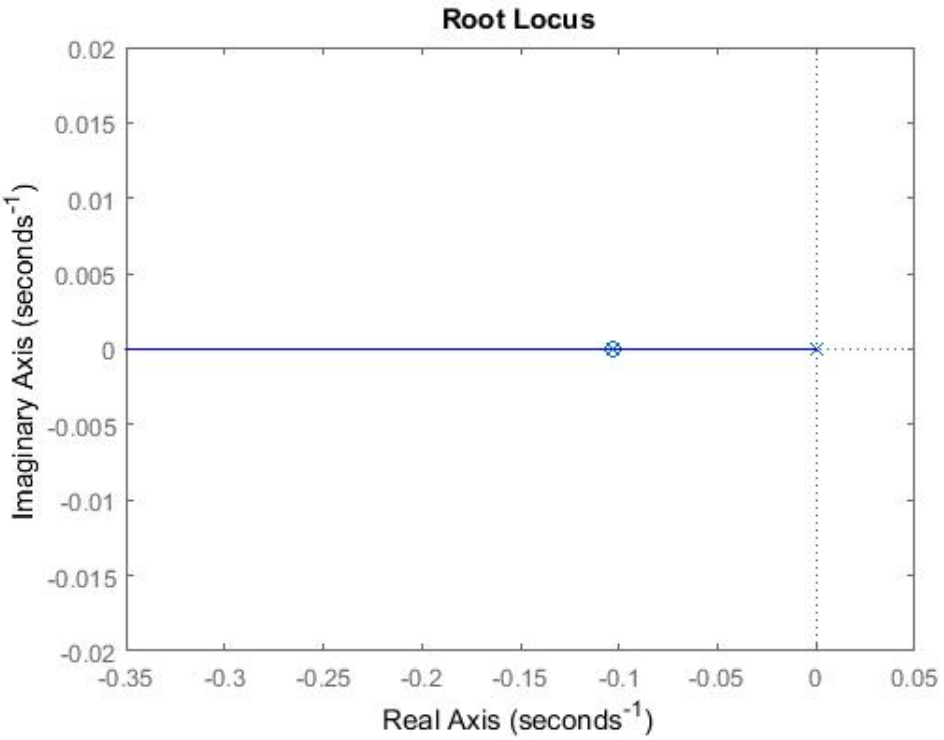

Figure 3.19: Velocity control root-locus

.

Analyzing the root-locus presented, it is possible to assert that the response to the step of the closed-loop system has no oscillation or overshoot since the closed-loop pole is always real. Therefore, this allows the tune of the settling time based on the value of the proportional gain. The higher value given to the proportional gain the faster the system will converge to the velocity reference but the higher current it will be required.

The PI controller law obtained is given by:

$$F(s) = K_p \frac{s + 0.1029}{s} E(s) \tag{3.36}$$

As an example, for a K_p equal to 10, the step response obtained is illustrated in figure 3.20.

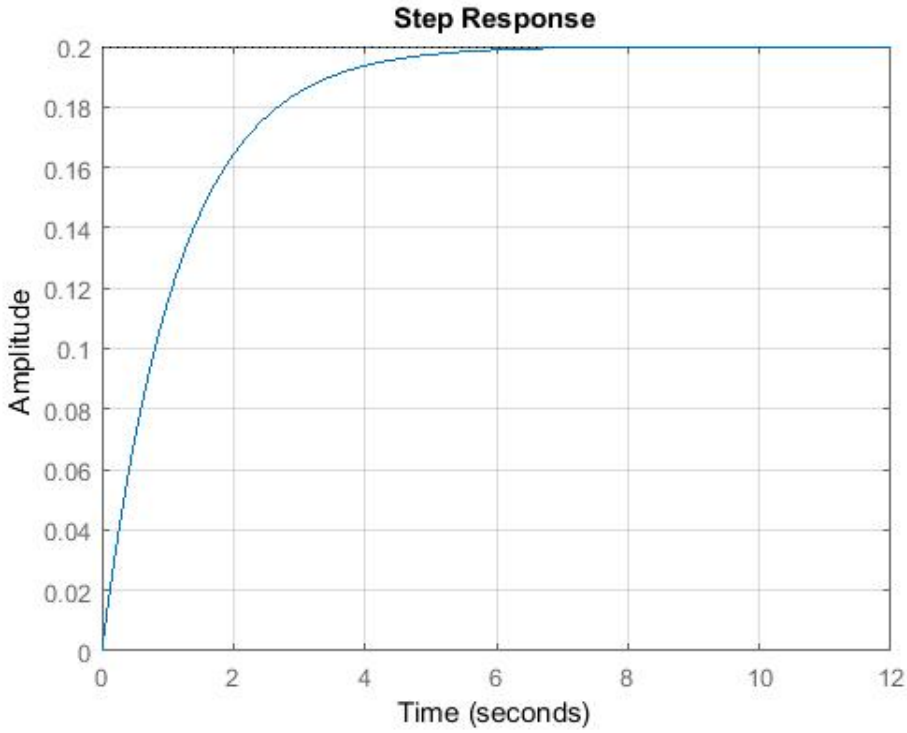

Figure 3.20: Velocity control response for K_p equal to 10

.

The design of the velocity controller is focused on making the controller parameters flexible towards the mission demands and take into account the vehicle's characteristics.

Considering that the velocity controller task is to approximate the vehicle towards the obstacle while it is still far away, it is considered that in most situation it will be the controller responsible for most of the navigation of the vehicle and therefore the main concern for its design is the energetic cost. The energetic cost depends on the power enforced by the controller and the time necessary to reach the switch point to the position controller and therefore this information is used in order to design the controller. Furthermore, the choice of the parameters also takes into account the maximum current enforced by the controller in order to guarantee that the thrusters can provide the necessary force.

A method was created composed by a series of steps capable of calibrating the controller's parameters based on the use of the simulink diagram presented in figure 3.21, allowing the choice of a controller that is suitable to each mission.

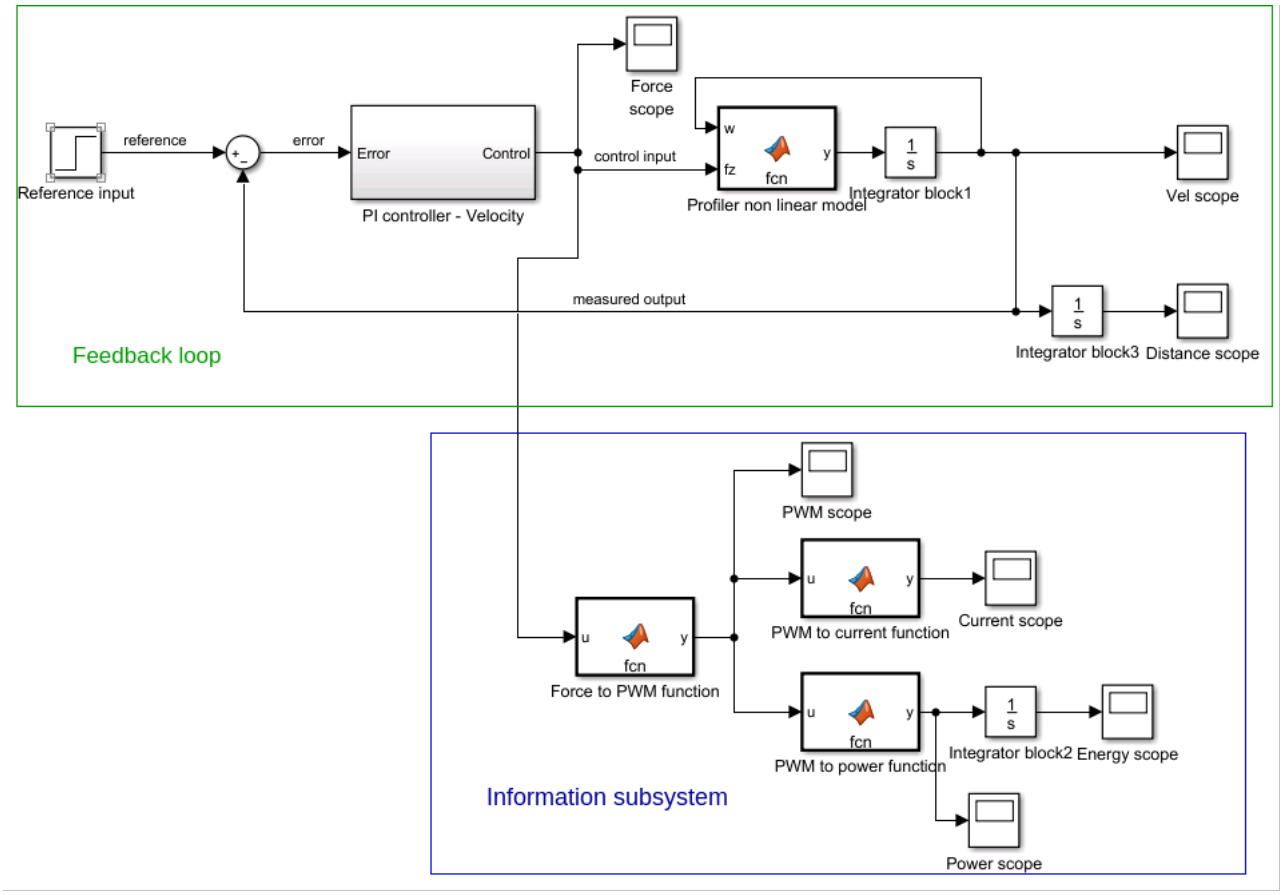

Figure 3.21: Simulink diagram for the velocity control design

.

The diagram contains a proposed controller dimensioned particularly for the profiler used in this dissertation. This controller was designed in order to require the maximum current that the thrusters can provide. The thrusters selected in [1] were the T100 thrusters by BlueRobotics [44]. The speed of the motor is controlled by four Afro Slim ESC20 Amp [45]. The overall setup, as stated in [1], allows a maximum current of 11.5 Ampere.

The controller was designed in order to require approximately the maximum current stated on the startup of the actuation. In order to design this controller, the relation on time domain between the actuation and the parameters of the controller and reference input was obtained through the following procedure:

As stated previously the PI controller law is given by:

$$F(s) = K_p \frac{s+Z}{s} E(s) \tag{3.37}$$

The relation between the derivative of the force actuated and the error on the time domain can be obtained by applying the Laplace inverse:

$$\dot{f}(t) = K_p \dot{e}(t) + K_p Z e(t) \tag{3.38}$$

It is necessary to calculate how the error evolves over the time. This can be done through the analysis of the close-loop response of the overall system.

Since the process transfer function is given by:

$$Process = \frac{K}{s+0.1029} \tag{3.39}$$

The close-loop behavior can be defined as follows:

$$Closeloop = \frac{K_p K}{s+K_p K} \tag{3.40}$$

Applying an step to the closeloop system, and through the inverse Laplace transform, the velocity behavior is given by:

$$Velocity(t) = V_{ref}(1-e^{-KK_p t}) \tag{3.41}$$

The error is therefore:

$$e(t) = V_{ref} - V_{ref}(1-e^{-KK_p t}) = V_{ref}e^{-KK_p t} \tag{3.42}$$

Substituting the $e(t)$ in the equation 3.48:

$$\dot{f}(t) = V_{ref}(-K_p^2 K e^{-K_p K t} + K_p Z e^{-K_p K t}) \tag{3.43}$$

Integrating the equation, it is possible to obtain the following relation:

$$f(t) = V_{ref}e^{-K_p K t}(K_p - \frac{Z}{K}) \tag{3.44}$$

Since the highest force required from the controller will always take place at the start of movement, substituting $t = 0$:

$$f(0) = V_{ref}(K_p - \frac{Z}{K}) \tag{3.45}$$

This equation allows the expression of the parameter K_p in relation with the V_{ref} and the maximum force:

$$K_p = \frac{f(0)}{V_{ref}} + \frac{Z}{K} \tag{3.46}$$

In the performance charts of the thrusters, it is possible to access the relation between the force required and the corresponding current and therefore using the following expressions it is possible to design a controller that guarantees an specific maximum current at the start up:

$$PWM = (0.2038 + sqrt(2.7e^{-4} - 8.2e^{-7}))/(1.36e^{-4}) \tag{3.47}$$

$$F_{kgf} = 7.2e^{-3} - 11.28 \tag{3.48}$$

For each mission, since the demands might change, it is possible to design the most adequate controller by applying the following steps using the simulink diagram in 3.21:

- First, the velocity reference of the velocity controller is defined. Based on this information it is calculated the switching point between the controllers as explained in 3.3.3.2. This indicates at what altitude the switching will take place. Knowing an approximation of the total depth of the water that the profiler will navigate, the distance that the profiler will have to travel during the velocity control is calculated;

- Knowing the maximum current that it is imposed in the mission, the maximum possible performance controller is designed through the procedures mentioned;

- After configuring the controller in the simulink diagram presented in 3.21 with the given parameters for the maximum performance, using the Distance scope and the distance that the profiler has to travel before the switching it is possible to calculate how much time it will take for the vehicle to reach that point;

- Through the Energy scope, the evolution of the energetic cost can be viewed. Using the information calculated in the last step it is possible to know the total cost of the navigation using the velocity control;

- From this point, it is necessary to decide if the total cost is acceptable or if the controller should be tuned down. The procedure to tune down can be made by reducing the gain value and repeating the steps here described until the desired controller is obtained.

The images 3.22 and 3.23 illustrate the transient response of the velocity (Vel scope) and the current (Current scope) for an input reference of 0.2 m/s and a controller designed to require the maximum possible current using the mentioned procedure (k = 13.5357):

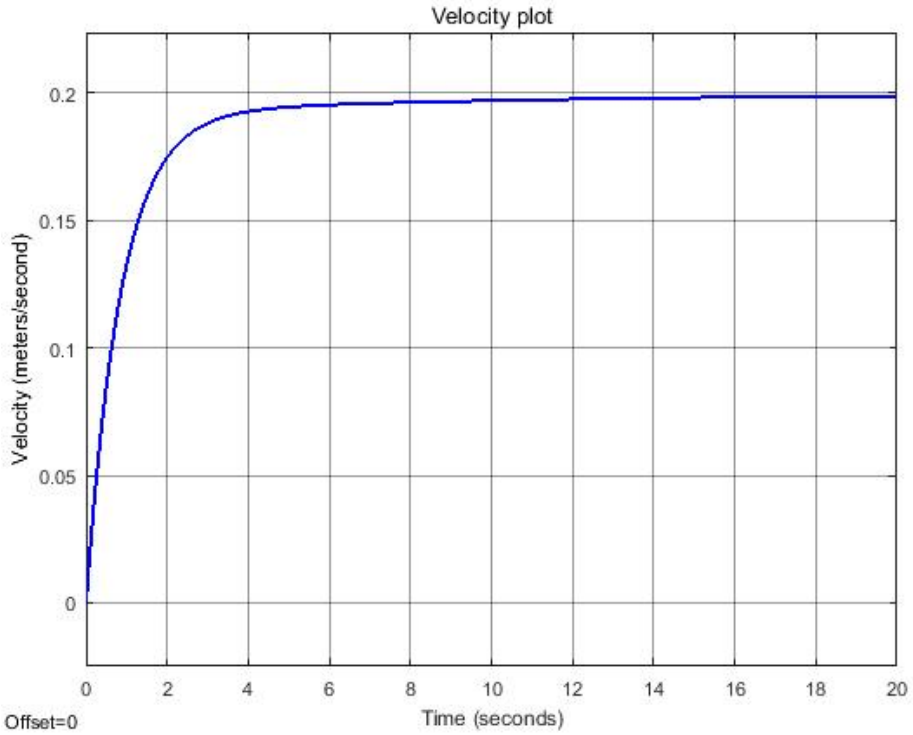

Figure 3.22: Transient response of the velocity

.

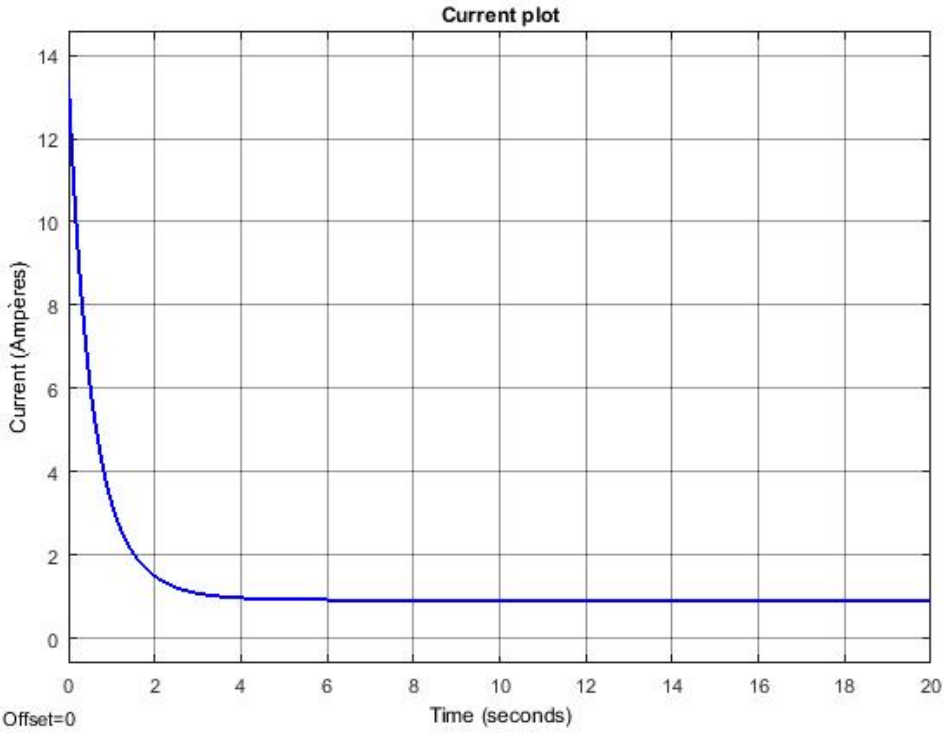

Figure 3.23: Current scope

.

Analyzing the velocity response plot 3.22, it is verified that since it is used the non-linear profiler mathematical mode, the presence of the flotation force slows the transient responde. It is also possible to assert that the use of an integrative term on the controller handled the flotation force, achieving the pretended velocity reference.

The figure 3.23 shows that the current imposed on the thrusters was a small amount greater than the maximum current that the controller was designed to impose. This is justified by the use of the linearized model in order to design the controllers instead of the non-linear model as used during the simulation. In a real application it is recommended that an safety margin is applied to the controller obtained for this reason.

Related to the discrete approximation of the controller, as stated in 3.3.3.2 it is expected that its behavior is identical to the continuous controller. The comparison between the response of the system using the continuous and the discrete for the example given in 3.20 is illustrated in figure 3.24.

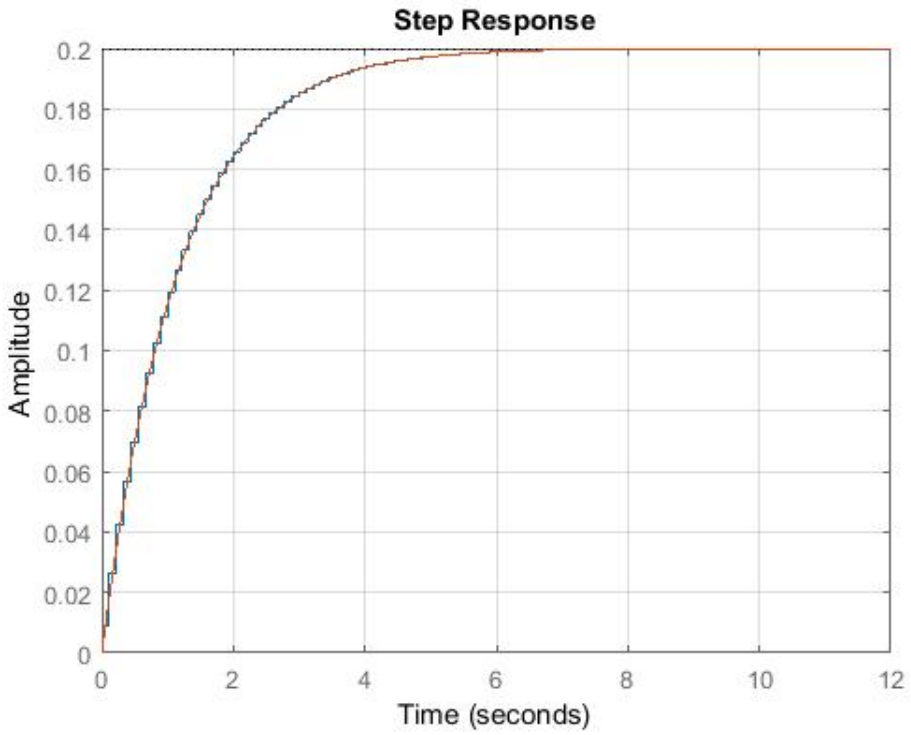

Figure 3.24: Comparison between the response of the system using the continuous and the discrete controllers for velocity

.

3.3.4 Integration of the two controllers and corresponding validation

In order to validate the behavior of the control module, specially to evaluate the switching between the 2 controllers, the simulink diagram present in figure 3.25 was created.

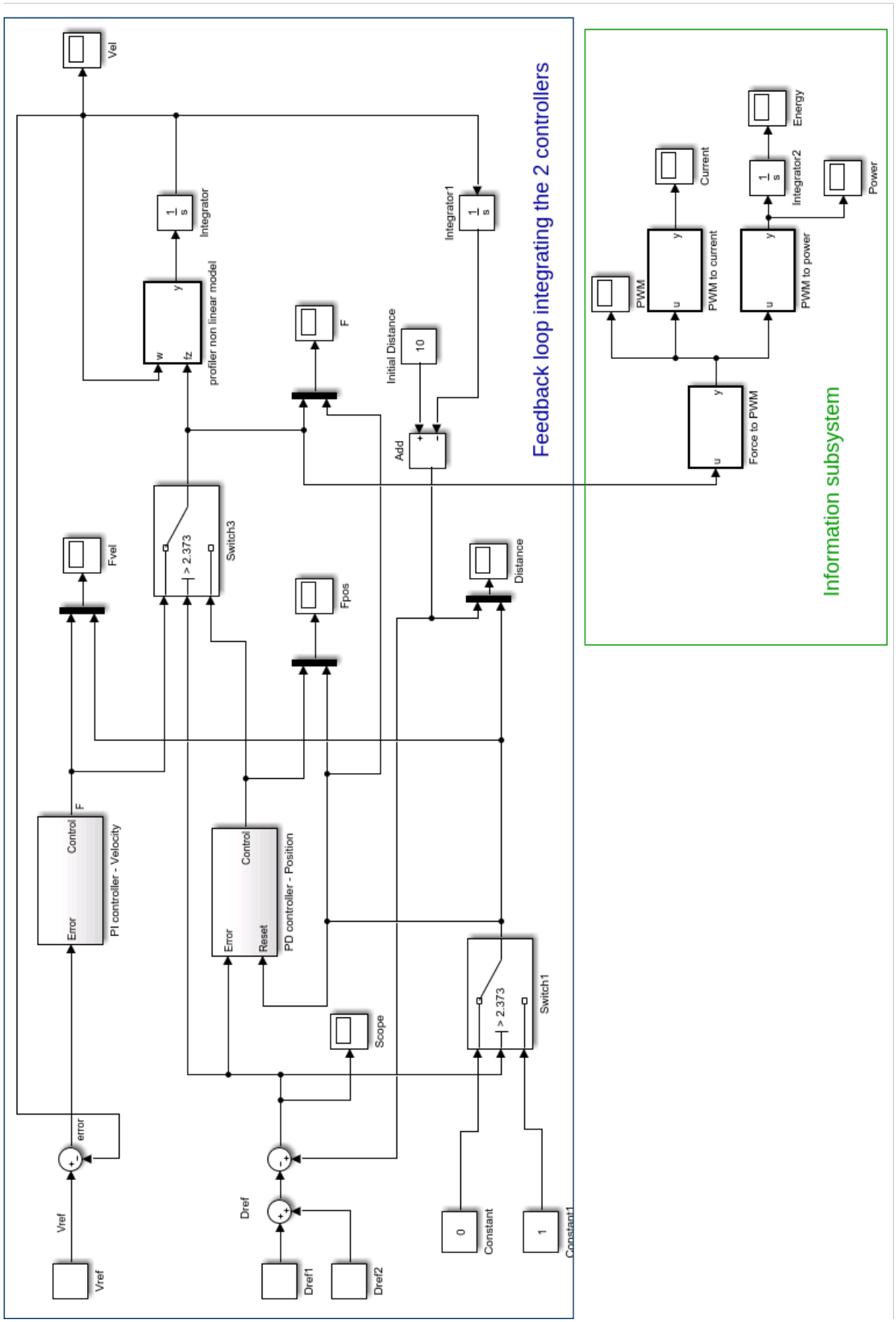

Figure 3.25: Simulink diagram to test both controllers working together

.

The diagram contains the position controller (PD controller - Position), the velocity controller (PI controller - Velocity), the profiler non-linear model using the equation 3.15, and the necessary switch blocks that activate in an already calculated value following the steps in 3.3.3.2.

Since the controllers were dimensioned using a linear model that does not include the effect of the flotation of the vehicle, it is necessary to introduce a solution that contracts this force. Relative to the velocity controller, it is not mandatory to take action referent to the flotation since it has an integral component that over time will negate the flotation force as shown previously.

In order to solve the problem that the flotation induces in the position controller, some solutions were considered. The 2 most relevant ones are the following:

- Addition of a force equal to the flotation force with in the moment of the transition that would persist during the control and conteract the flotation;

- An solution based on feed-forward that would increment an additional force based on the response of the system in order to negate the flotation force.

The first solution is the simplest but would require the knowledge of the flotation force before the design of the controllers. The second solution would require the tuning of the pace of addition of the force that could be extremely limited due to the necessity to guarantee that the change on the actuation of the motor is smooth. This could lead to a considerable slow down of the system response.

Considering the two solutions presented, it was chosen the addition of a force equal to the flotation at the switching moment. This solution requires the execution of some tests in order to have an estimation of the flotation force. Furthermore, the addition of this force causes a spike on the actuation of the thrusters, which can be solved by considering that the force necessary at the switch to have into account this increment. Therefore, the equation mentioned in 3.34 is changed to the following equation:

$$Error = \frac{F_{required} - f_{flot} - K_d V_{ref}}{K_p} \tag{3.50}$$

where f_{flot} represents the value obtained for the flotation force by the experiments a priori to the mission.

Using the diagram presented in figure 3.25, for a velocity controller with K = 10, the position controller shown in 3.3.3.2, velocity reference of 0.2 m/s and position reference of 0.1m of altitude the results obtained are presented in figures 3.26 to 3.31.

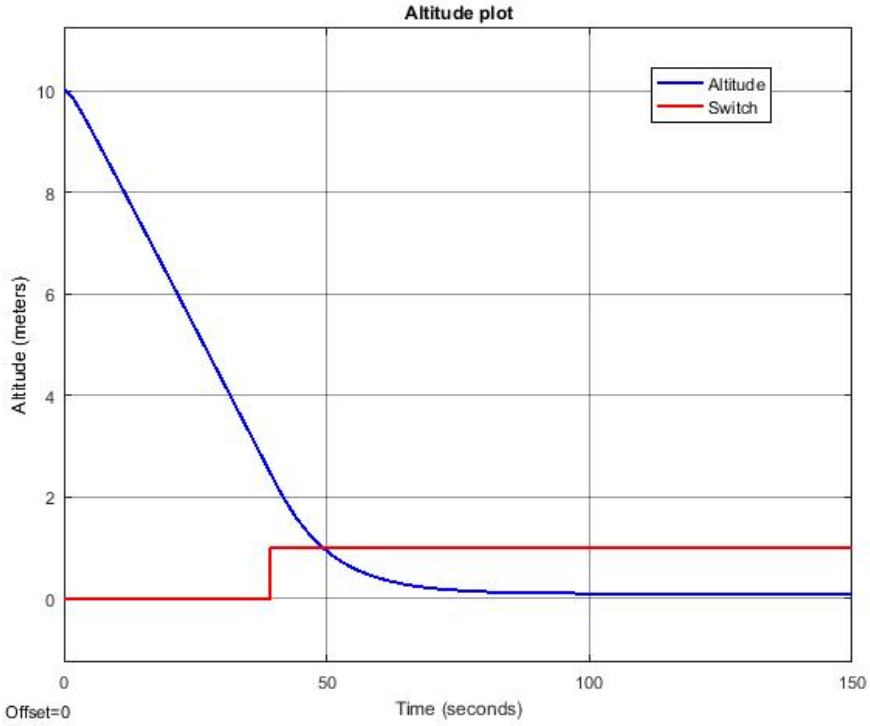

Figure 3.26: Altitude scope

.

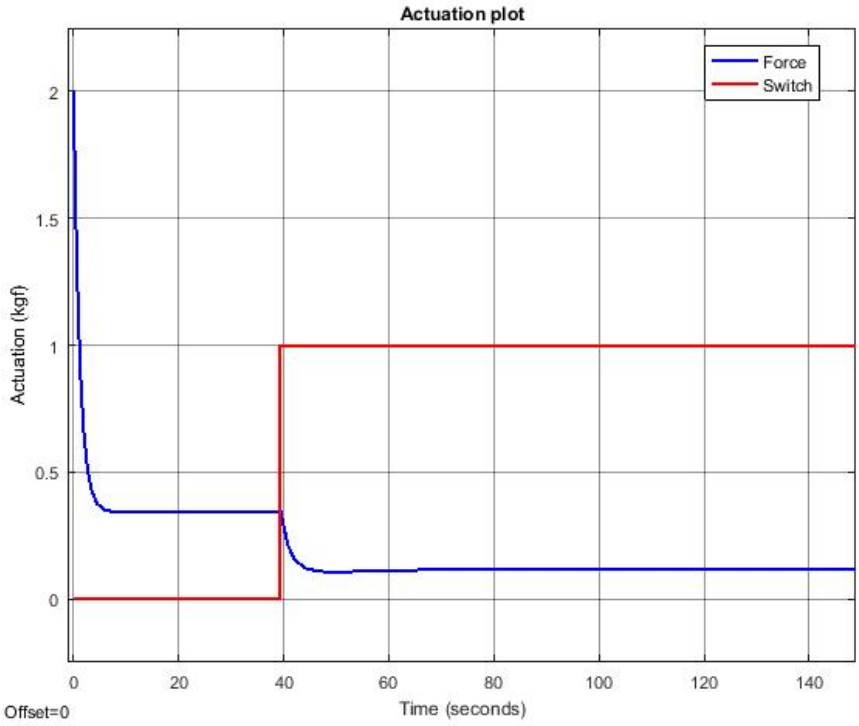

Figure 3.27: Actuation scope

.

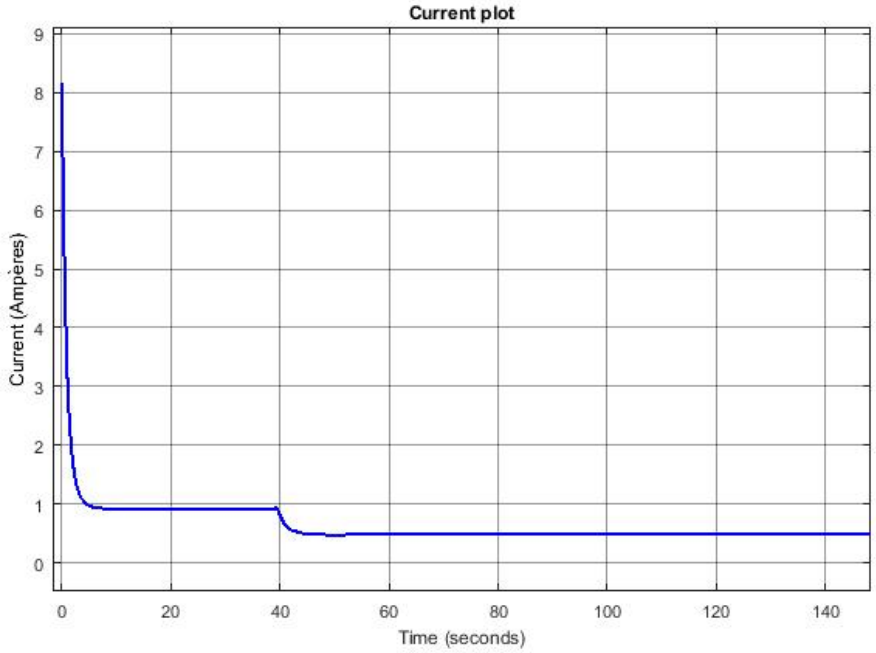

Figure 3.28: Current scope

.

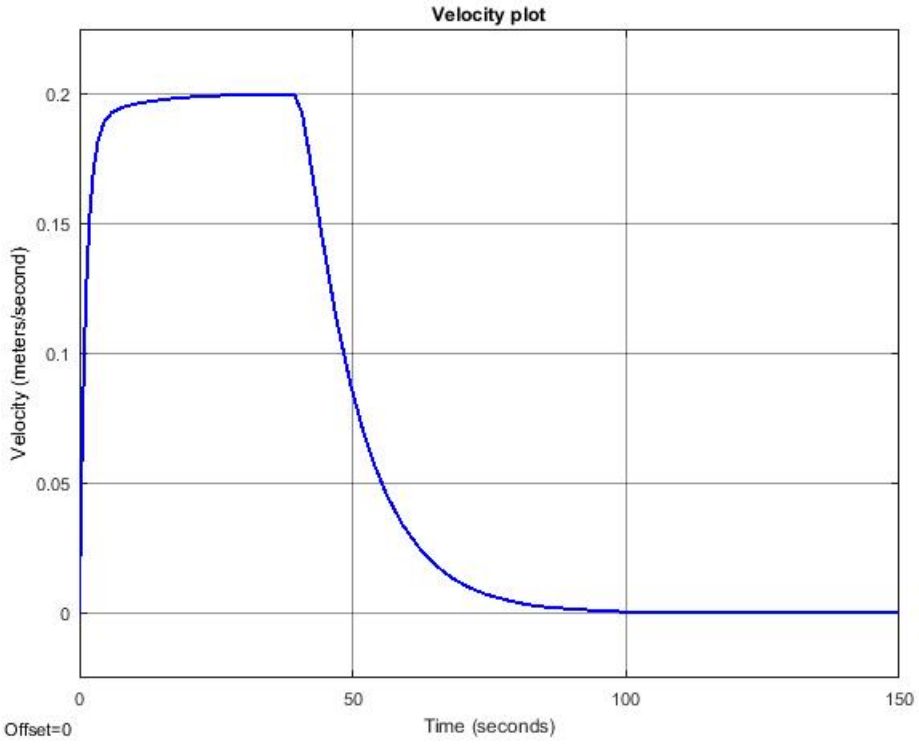

Figure 3.29: Velocity scope

.

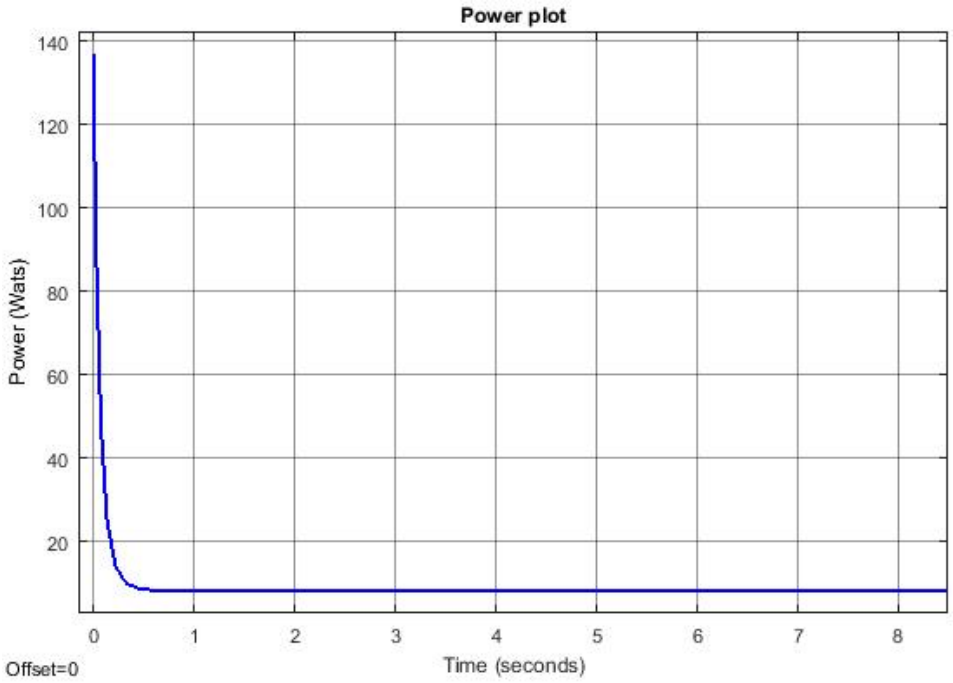

Figure 3.30: Power consumption scope

.

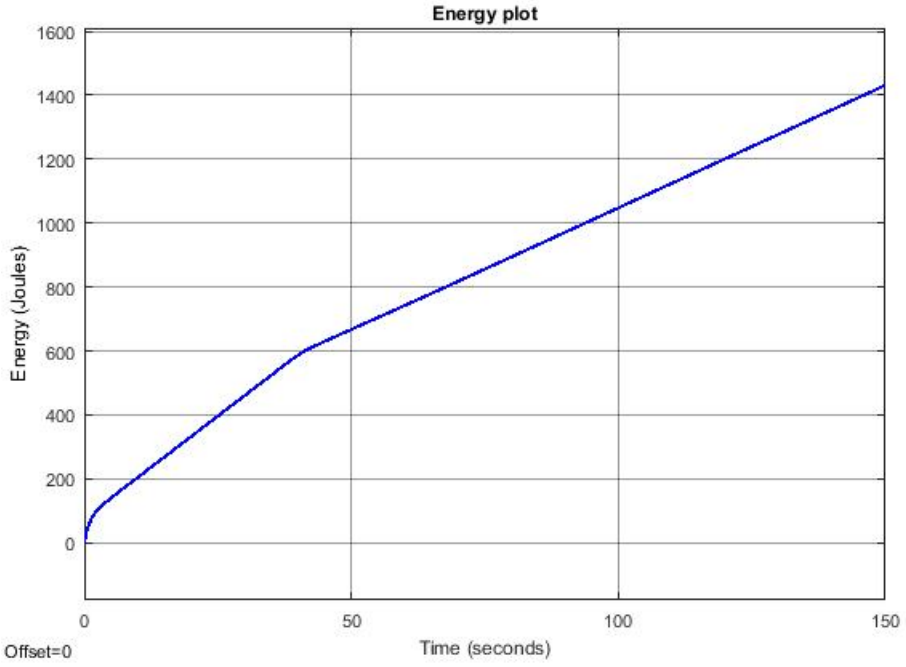

Figure 3.31: Energetic cost scope

.

Analyzing the images presented it is possible to assert that the switching between the controller was done smoothly as pretended.

Relative to the energetic cost of this mission, the vehicle reached the position reference after approximately 110 s, using the scope presented in figure 3.31, the total energy cost was approximately 1050 J (0.292 Whr). The batteries present in the profiler are 2 Ocean Server BA95HC-FL batteries [1], yielding a total capacity of 190 Whr. The amount of energy taken in this movement corresponds to only 0.154% of the total capacity of the batteries.

It is also possible to observe that the transient response of the position controller was not as fast as the designed one. This is justified by the fact that the position controller was enabled during the full movement making the derivative factor not as high as if the controller was activated in the switching instance. This was done in order to prevent a spike on the switching point since the derivative factor would be extremely high when the position controller activated. The position controller being enable during the full movement is the same as using the variation of the altitude to feed the derivative instead of the error, allowing the prevention of spikes on the actuation when the reference changes.

Another factor that is necessary to take into account is the fact that it is considered that the velocity reaches the reference before the switching point as stated previously. In a situation where the velocity does not reach the reference, the actuation of the position controller will be different when compared to the velocity one on the switch point and might provoke a spike on the actuation. To prevent this, a limiter of the variation of the actuation can be added.

The figure 3.32 illustrates the situation of an insertion of a new reference equal to 0.5 *meters* after 150s.

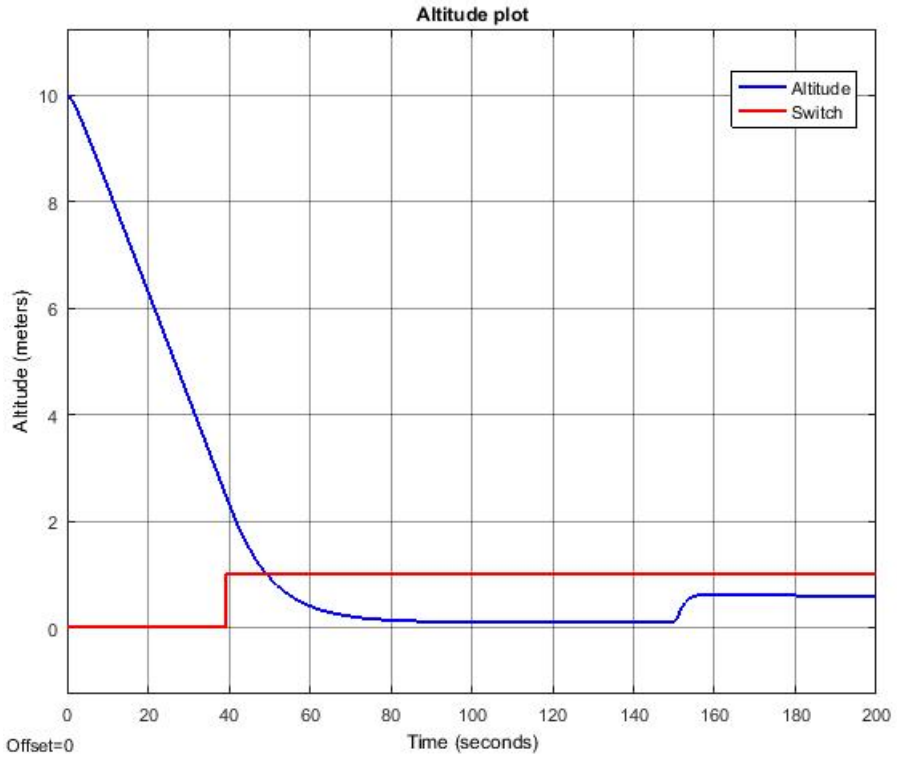

Figure 3.32: Altitude scope with 2 references

.

It is possible to notice that the response to the second reference in this simulation is as it was designed, being different from the response to the first reference. This is justified by the fact that in this second reference it was not used the change of the altitude as the input of the derivative term of the controller. The simulink toolbox does not allow the use of the change of the altitude as the input of the derivative and therefore it is expected a peek in the force and consequently, in the current when the new position reference is given. This problem is not relevant in a real environment where the input of the derivative term can easily be controlled.

Chapter 4

Integration and evaluation of the modules

Using the simulink toolbox it was possible to visualize the behavior of the control system when the system gathers the altitude and velocity of the vehicle from a "perfect sensor". It is necessary to validate the behavior of the system when the 2 modules developed are integrated and therefore it is used the Sensor module to gather the information that feeds the control. This can be achieved using a simulation environment for underwater vehicles and posterior integration of the modules in the profiler, testing its performance in a real environment.

The use of an underwater simulation environment allows a high controllability and observability over the tests. Through this simulation it is possible to simulate a vast number of environments and conditions to predict the behavior of the system in a real environment. After the simulation process, the experimental tests using the vehicle allows the test of the profiler's mathematical model and allows the evaluation of the behavior of the system in a real environment where multiple other problems are present.

Since the profiler was not available for tests during the time of this dissertation due to problems not related to this dissertation, it was not possible to evaluate the system when both modules were integrated in the profiler. The simulation using an underwater simulation environment was reinforced and adapted in order to better simulate the architecture that would be working in the profiler after the integration of both modules. The procedures and simulation results from the use of the underwater simulation environment is presented in the following section.

4.1 Underwater simulation environment

It was used the UWsim simulation environment [46] that allows the simulation of the movement of an underwater vehicle in an underwater virtual scenario.

The objectives of this simulation are the following:

- Validation of the overall architecture developed that integrates both modules in the profiler's system;

- Validation of the controllers in discrete time;
- Integration of the sensor module and the controller module, and validation of the control of the profiler using the measurements from the sensor module;
- Evaluation of the Kalman filter, especially in situations where the Sensor module could not acquire a reliable measurement and the pressure sensor must be used;
- Visualization of the movement of the vehicle in an underwater virtual scenario in order to better evaluate the controllers.

In order to simulate the system using the UWsim, the example Girona500 given in the tutorials of the UWsim [47] was used. Several changes were made in the files presented in this example to implement the profiler characteristics, the controllers designed, and to create an underwater virtual scenario that allows the objectives mentioned.

The main 4 files that were created or adapted are the following:

- dynamicsAUP - Python file based on the Girona500 example that implements the dynamics of the vehicle;
- dynamicsConf - File containing the dynamic model parameters;
- aupCamera - XML file that determines the underwater virtual scenario.
- ThrusterControl - Python file that is responsible for the application of the speed and position controller;

The main difference made in the dynamicsAUP file in comparison to the Girona500 example is the determination of the flotation force that was adapted in order to consider the drag coefficient and the flotation force mentioned in 3.3.3.1.

The dynamicsConf format is the same as the one presented in the Girona500 example, where the values were changed according to the parameters obtained for the model of the profiler.

The aupCamera file was created based on the conventional underwater virtual scenarios presented on the tutorials of the UWsim [47], and it contains a virtual camera in order to gather images simulating the acquisition of the Sensor module, a light projector to simulate the laser pointer device, a pressure sensor, and a sonar device. The overall scenario simulated is a typical ocean-like environment with a box object near the bottom of the ocean in order to guarantee that the surface is flat and more controllable. Figure 4.1 illustrates the underwater environment used.

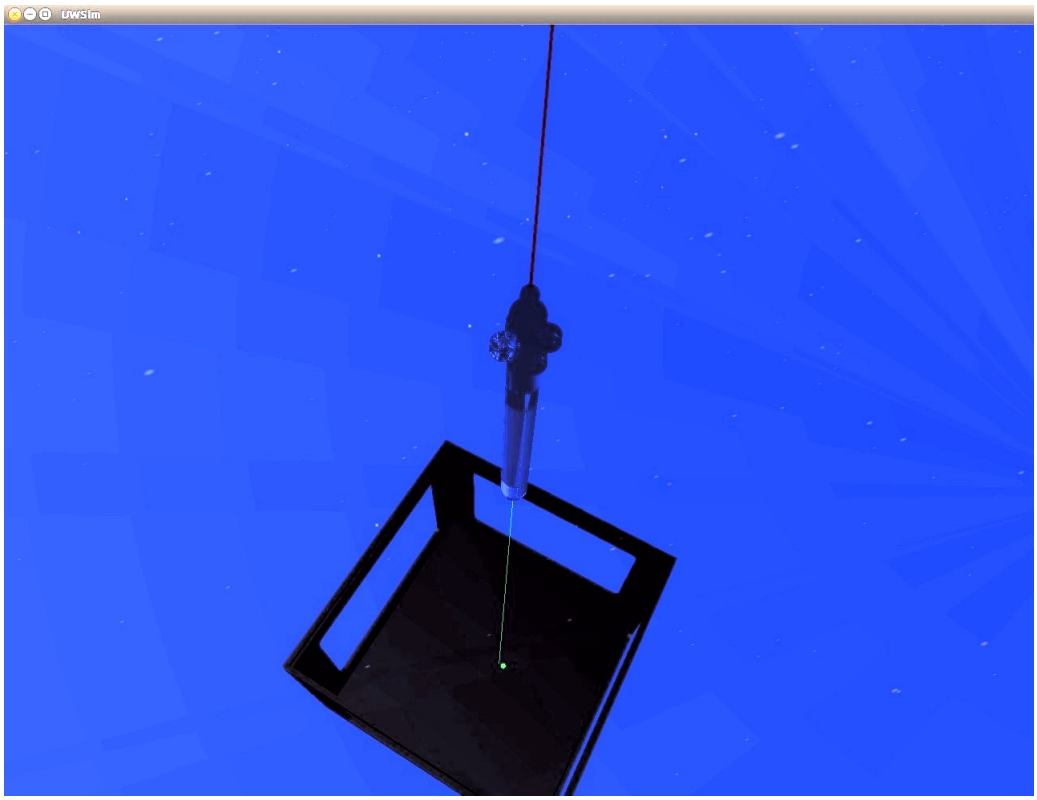

Figure 4.1: Underwater scenario simulated by UWsim

The ThrusterControl file contains a control cycle where the controllers designed were implemented in order to determine the actuation on the thrusters.

Furthermore it was developed a ROS workspace referent to the Sensor module in order to make possible the interaction between the UWsim and the Sensor module software. It contains the following 3 main components:

- A main file responsible for the interaction between the Sensor module and the UWsim;
- The imageProcessing class developed;
- The KalmanFilter class developed.

The main file is in charge of the following main tasks:

- Subscription to the relevant topics that retrieve needed information (a topic that contains the images that are taken periodically by the virtual camera, a topic containing the measure of the pressure sensor, and a topic that publishes the position of the vehicle using the dynamic model);
- Configuration of the imageProcessing and dispatch of the images received to the imageProcessing class procedures;

- Creation of a shared memory that contains the value of the actual altitude, pressure, and position using the dynamic model.

The imageProcessing class is essentially the same class as the one present in the Sensor module. It receives the image dispatched by the main file of this ROS workspace, processes it, sends the altitude to the Kalman Filter and then shares the final state using the shared memory block. The only significant difference is that when it sends the altitude to the Kalman filter, it also accesses the shared memory in order to acquire the pressure value, sending also this value to the Kalman filter.

The Kalman Filter receives the altitude calculated and the value of the pressure presented in the shared memory block and implements the filter explained in 3.3.2.

The overall architecture was developed in order to make this simulation as close as possible to the real operation of these modules after their integration in the profiler.

The figure 4.2 illustrates the architecture and the typical sequence that runs in the simulation.

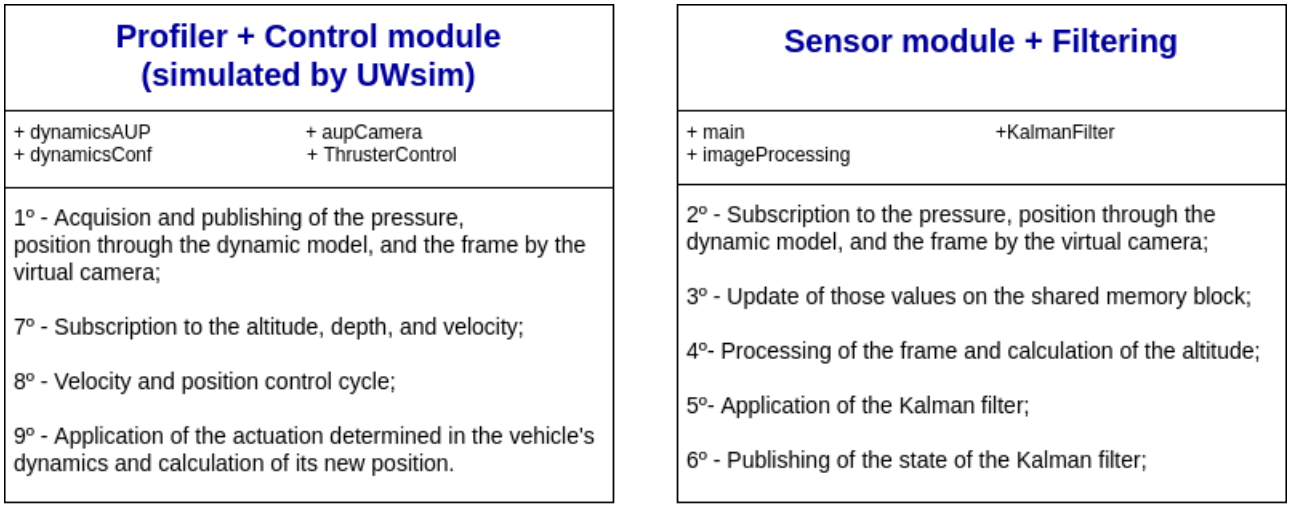

Figure 4.2: Overall architecture and sequence of actions that take place in the simulation

4.2 Tests performed

The first objective of the simulation was the validation of the overall architecture. Furthermore, this initial test also allows the validation of the discrete controllers in this underwater environment, the usage of the sensor module, access to the shared memory block, and the evaluation of the actuation of the Kalman filter in situations where the sensor module could not retrieve measurement and therefore it is necessary to resort to another sensor.

In order to do so, the following conditions were used:

- Starting point at approximately 4 m of altitude;
- Position controller with an input reference of 0.3 m and the following parameters, K_p = 0.5145 and K_d = 5, as designed in 3.3.3.2;
- Velocity controller with an input reference of 0.1 m/s and parameters K_p = 10, K_i = 1.0290, designed using the procedures mentioned in 3.3.3.3;
- Switching at 1.1687 (Switching error point) + 0.3 (position reference) m, calculated as stated in 3.3.3.2;
- Consideration that the measurements of the Sensor module when the vehicle is at an altitude higher than 2.5 m are not reliable;
- Usage of the z position given by the dynamic instead of the pressure sensor simulated by the UWsim in order to have more control over the error present in those measurements, guaranteeing that for this test the sensor used is considered a "perfect" sensor.
- The frequency of frames given by the simulated camera is 9 fps in order to better simulate the camera used by the profiler.
- The period of the control cycle is 100ms;

Relative to the Kalman filter, the following matrix were considered for the matrix of the error induced related to the process (Q) and for the matrix representative of the error in the measurements (R):

$$Q = \begin{bmatrix} 0.05 & 0 & 0 \\ 0 & 3e^{-6} & 0 \\ 0 & 0 & 3e^{-6} \end{bmatrix}$$

$$R = \begin{bmatrix} sensorError & 0 & 0 \\ 0 & (10^{-7})^2 & 0 \\ 0 & 0 & 2 \times (10^{-7})^2 \end{bmatrix}$$

The values of the matrix Q were determined empirically in order to reduce the abrupt changes that might happen on the measurements and therefore in the estimations.

The matrix R contains the error of the sensor module ($sensorError$) that depends on the current altitude as shown in 3.2.8. It also contains the error of the sensor that is responsible for the depth measurement. Since in this test it it was used the value of z of the dynamic model as stated to obtain the depth value, it is considered that it is used a "perfect sensor", justifying the low value of 10^{-7}. The error relative to the velocity is 2 times greater than the error for the depth since the velocity is calculated based on an arithmetic operation between 2 values of depth, and therefore it can be asserted that it contains 2 times its error.

The results obtained are presented in the following images:

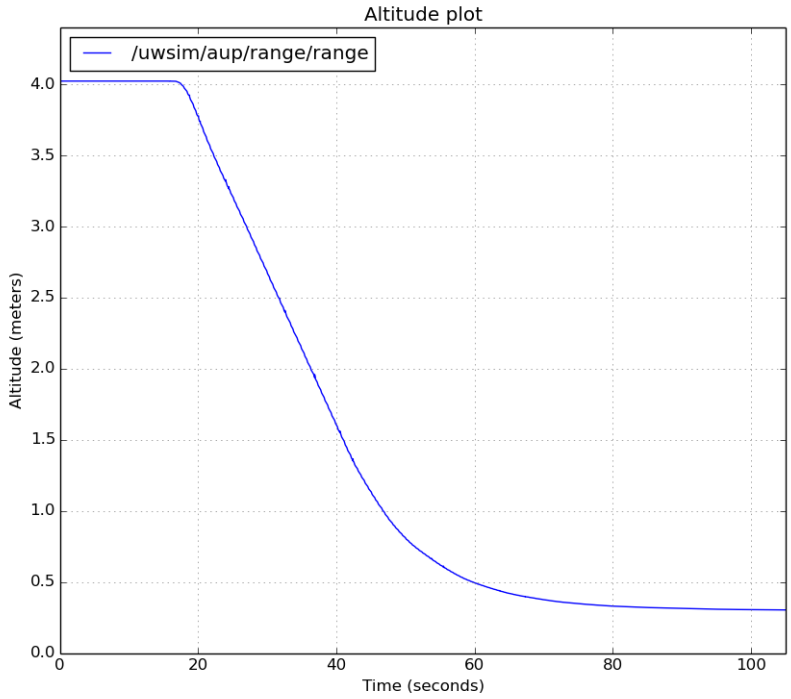

Figure 4.3: Altitude plot

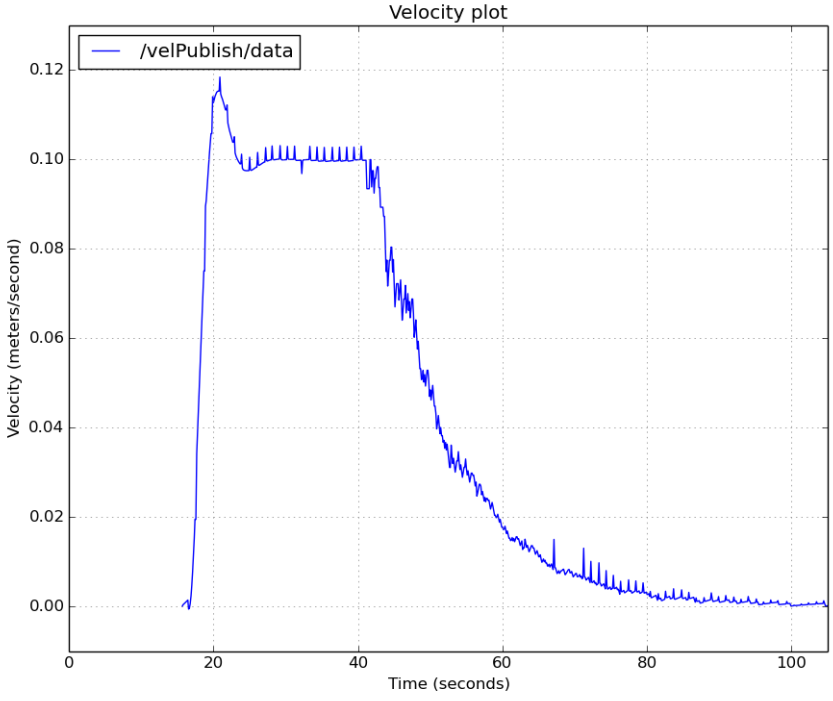

Figure 4.4: Velocity plot during the velocity control

The plot 4.3 was gathered using the simulated sonar sensor present in the simulated underwater scenario in order to have the information about the real altitude throughout the whole mission. This

test proved that the architecture worked as designed, and the Kalman filter is able to use the depth value whenever the sensor module can not gather viable measurements.

Relative to the discrete controllers' performance it is possible to observe that the position controller worked as intended as seen in figure 4.3. Relative to the velocity control, in figure 4.4 an overshoot is present that is not intended based on the design of the velocity controller. This is justified by the delay that the simulation and the Kalman filter induces in the overall performance. Its consequences do not result in a degradation of the quality of the movement of the vehicle and do not put the vehicle at risk and therefore are considered acceptable for this simulation purpose.

The next test is focused on the evaluation of the performance of the Kalman filter. In order to do so, the following conditions were used:

- Same starting point, controllers' parameters, switching point as the last test;
- Same consideration of the sensor module, where the measurements when the vehicle at an altitude higher than 2.5 m are considered not reliable;
- Usage of the z position given by the dynamic containing a Gaussian error induced;
- Introduction of forced errors on the sensor module measurements.

The reason for the use of the z position given by the dynamic containing a Gaussian error induced instead of the pressure sensor simulated by the UWsim is the inherent error that this sensor has. The pressure sensor simulated by the UWsim has approximately 3 cm of sensitivity that can not be altered. Therefore it was used the z position given by the dynamic containing an error approximately equal to the error of the pressure sensor that is used by the profiler. The pressure sensor used by the profiler is the Blue Robotics Bar30 [48] and contains 0.2 $mBar$ (equivalent to 2mm) resolution. The value of this sensor is transformed into depth considering that the environment has fresh water.

The values considered for the matrices of the Kalman filter are as follows:

$$Q = \begin{bmatrix} 0.05 & 0 & 0 \\ 0 & 3e^{-6} & 0 \\ 0 & 0 & 3e^{-6} \end{bmatrix}$$

$$R = \begin{bmatrix} sensorError & 0 & 0 \\ 0 & 0.002^2 & 0 \\ 0 & 0 & 2 \times 0.002^2 \end{bmatrix}$$

The forced errors introduced are the following:

- Error introduced when the vehicle reaches 1.5 m of altitude where the sensor module produces a high deviation measurement of 2 m during 5 frames (approximately 0.5 seconds)
- Error introduced when the vehicle reaches 0.5 m of altitude where the sensor module during 5 frames is not able to produce any valuable information; (approximately 0.5 seconds)

The errors introduced are meant to simulate situations where for example a particle or object circulates in front of the camera which can cause an incorrect detection, or cause the calculation of an incorrect measurement.

The figures 4.5 to 4.9 illustrate the results obtained.

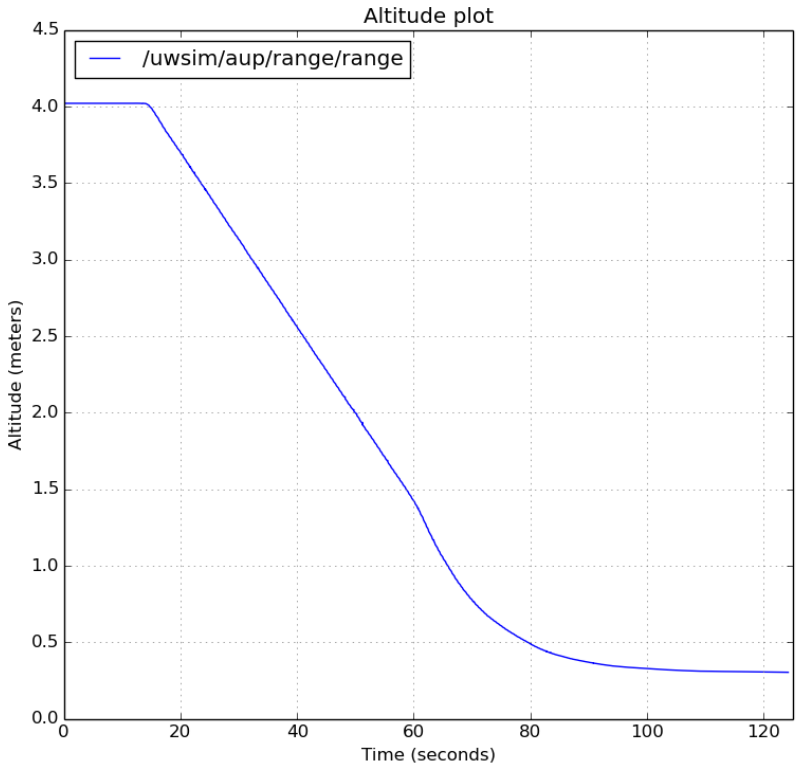

Figure 4.5: Altitude plot

The figure 4.5 shows that the overall intended behavior was achieved using the conditions mentioned. These conditions approximate this simulation to the behavior of the profiler navigating in a real environment where the use of sensors with an inherent error is unavoidable.

In the figures 4.6 and 4.7 it is possible to notice the errors induced. The error that causes the sensor to retrieve a high deviation measurement for 5 cycles is present at approximately 63 *s* and the error causing the measurements to not be reliable for 5 cycles occurs at approximately 81 *s*. It is possible to assert that the Kalman filter eliminated the errors induced by comparing the 2 figures.

In figure 4.8 it is shown a zoomed image showing the filtering of the Kalman filter in the depth measurements gathered. This shows not only the effect of the estimation of the filter but the pretended smoothing of the measurements to reduce the abrupt changes on the reference of the controller after the control cycle.

The figure 4.9 shows that the velocity reference was reached using the non "perfect" pressure sensor during the velocity control. The Kalman filter was able to adapt the values obtained leading to the correct control of the velocity and posterior position control.

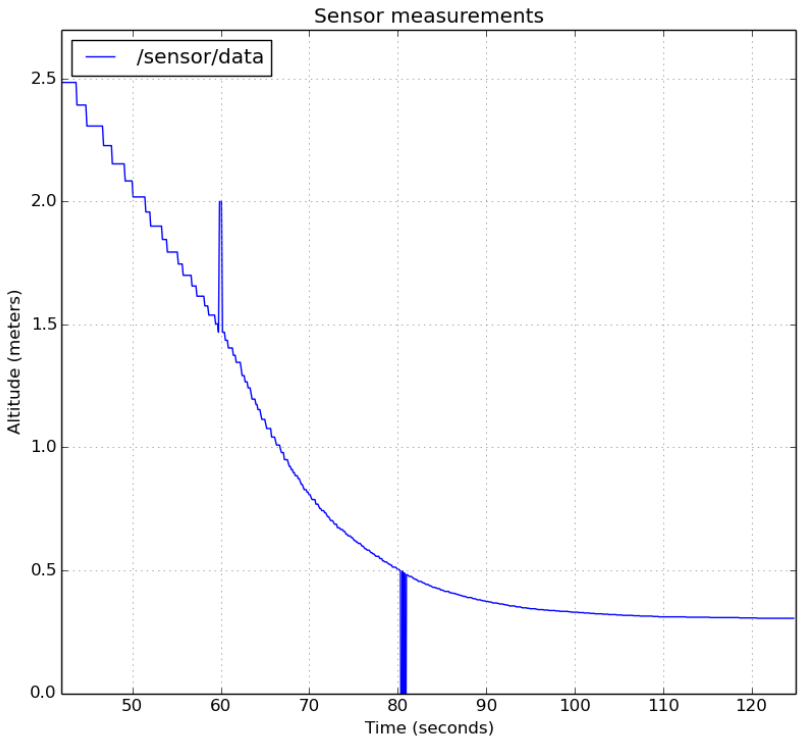

Figure 4.6: Measurements gathered by the sensor module after 2.5 *m*

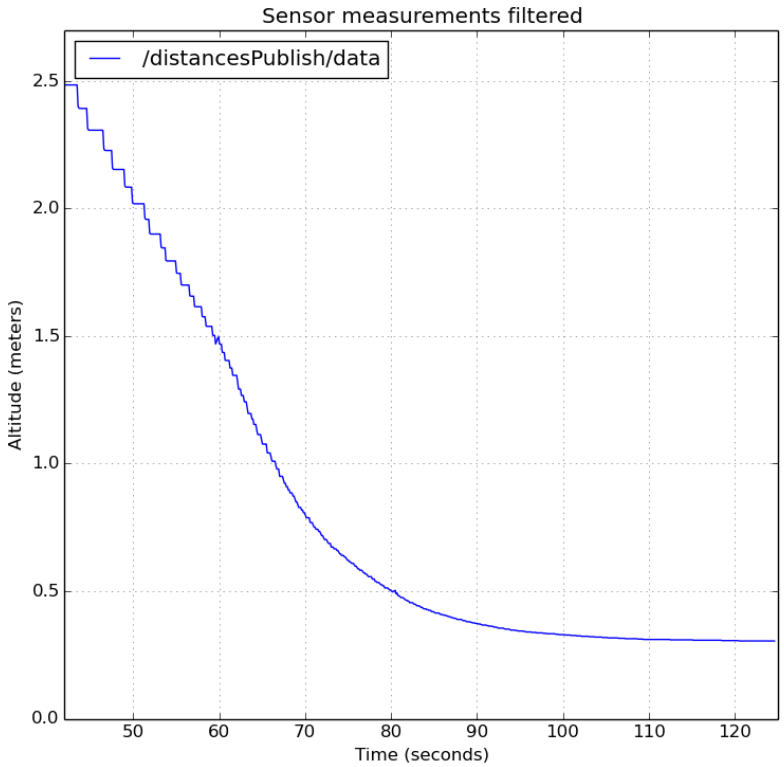

Figure 4.7: Filtered measurements from the sensor module after 2.5 *m*

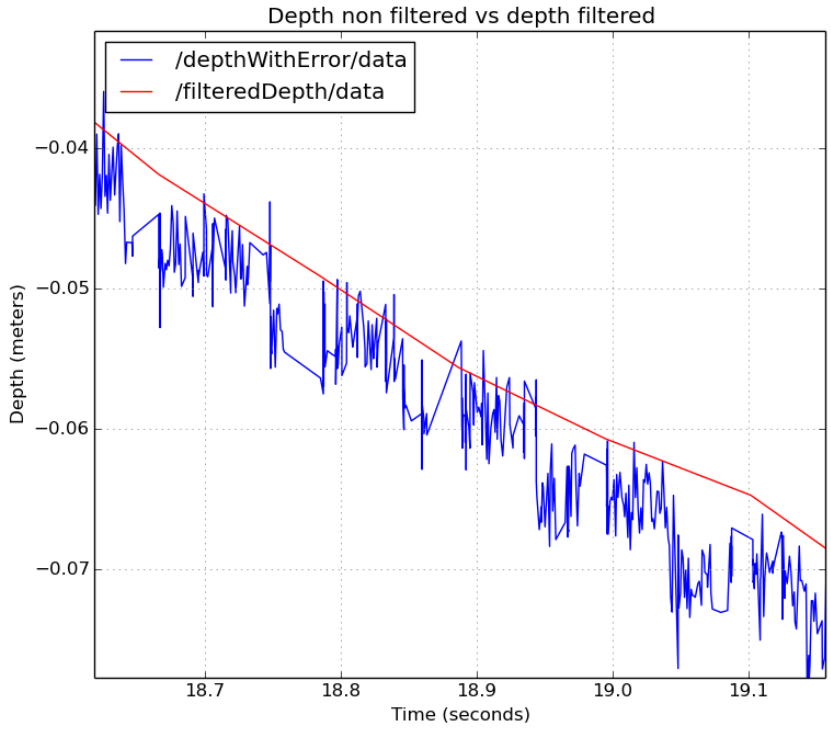

Figure 4.8: Depth measurements filtered vs non filtered

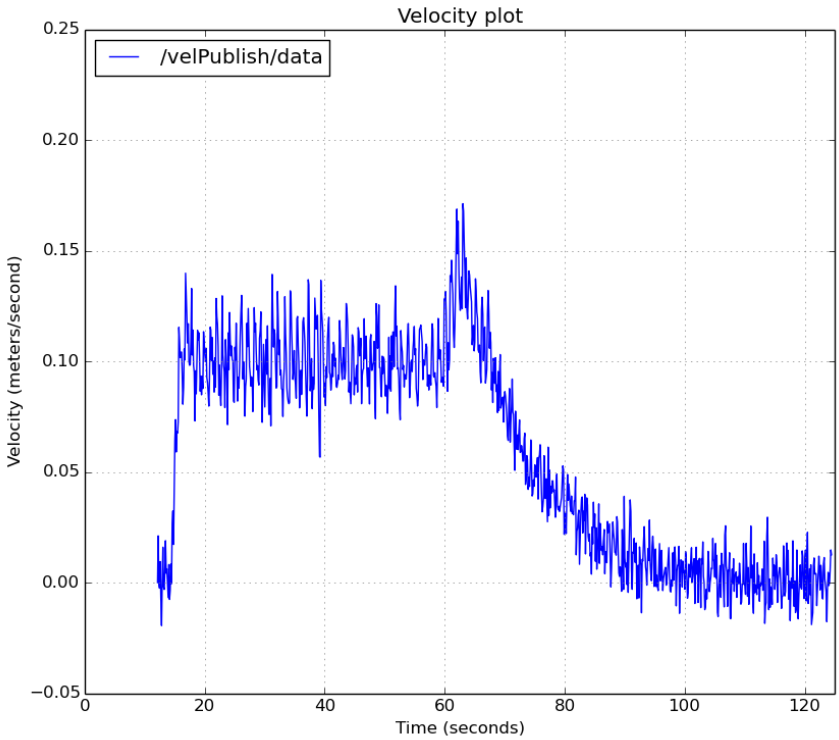

Figure 4.9: Estimated velocity plot

Chapter 5

Conclusions and future work

This chapter will present an overview of the work done as well as the conclusions drawn from it. Additionally, it will also provide some suggestions on possible future work that would benefit from work done in this dissertation.

5.1 Main contributions

The main goal of this dissertation was the development of a system capable of controlling the altitude of an underwater vehicle using computer vision. The work done was focused on the validation of the requirements presented in the chapter 3. In order to achieve this system, 2 modules were developed allowing the implementation of the two main inherent features (sensing and control).

The module responsible for the measuring of the distance based on computer vision (Sensor module) was implemented using 2 laser pointer devices placed parallel to one another beside a CCD camera. The software developed implemented the following features: Acquisition of frames; detection of the laser dots on the images; calculation of the centroids coordinates; calculation of the quality of the detection; calculation of the distance using the triangulation principle; data handling. The features were able to be implemented in a way that each can be improved without impacting the others. Furthermore, it was possible to include an error detection system in each phase to detect possible incorrect detections or calculations.

The triangulation principle revealed a considerable error inherent to its technology, limiting the range of altitude where it is beneficial to use this sensor module. It was verified that the error is inversely proportional to the camera resolution since the error is related to the pixel that contains the laser dot. It was also noticed that the error on the measurements is highly dependent on the calibration, especially the calibration of the angle of the camera and the laser devices. Overall, the use of image processing was successfully in order to gather the distance to a target, showing an error of less than 4 % in the range from 0 to 2.5 m, and an average relative error after the application of an average operation between the 2 measurements equal to 1%.

The module responsible for the implementation of the filtering phase and the control phase (Filtering and Control module) was developed using a Kalman filter with 3 state variables, a velocity controller, and a position controller. The use of two phases of control (velocity and position) and the use of a switching point revealed itself very effective and useful in order to be able to tune the controllers' parameters independently, allowing a better control on the overall movement. It was given special attention to allowing flexibility on the controllers' parameters depending on thesy mission's demands. The methods of tuning the controllers' parameters were successfully in determining a good relationship between the mission and vehicle's demands and the smoothness of the switching between the controllers.

The interaction between the 2 modules was designed in order to be able to integrate the profiler. The test of this architecture was made using the underwater simulated environmental UWsim, that showed good results on the interaction between the modules.

Furthermore, the tests using the UWSim allowed testing the control of the vehicle using the measurements of the sensor module filtered by the Kalman filter. The results obtained showed that the filtering phase was able to fuse the information between the sensor module and the pressure sensor in order to guarantee the control of the vehicle even in a situation where the sensor module could not retrieve any measurements. It was used 2 different pressure sensors with different inherent errors in order to show the impact on the overall control of the system. The results showed that the control was successful, being able to induce a good transition response on the vehicle altitude towards the reference with no steady state error.

Based on the results given for each module and for the tests made with the underwater simulated environmental using UWsim, it is possible to conclude that the main objective of this dissertation was achieved, and most of the defined requirements were accomplished.

An extended abstract of the state of the work done in this dissertation was submitted in April 18^{th}, 2018 to the OCEANS 2018 Charleston conference that will take place in October 22-25,2018. This conference main goal is the gathering of experts in the fields of marine technology engineering in order to reveal their work regarding the latest researches and innovation. The extended abstract document is present in A.

5.2 Future work

The system developed, despite accomplishing the main objective of this dissertation, still needs further testing. Initially, it would be beneficial to explore the simulation using the UWsim simulated environment in order to evaluate the system behavior using a diversity of conditions. For example, it would be beneficial to include a pitch, roll and yaw controller and demand the traveling of the profiler with a non zero angle related to the Z axis. This would represent a situation where the variation of the depth was not equal to the variation on the altitude and therefore the absence of measurements from the sensor module would have a bigger impact. Using this pitch, roll and yaw controller, it would be also beneficial to include an irregular seafloor combined with an induced current, verifying if the system could maintain the position reference along the seafloor.

After the simulation tests, it would be opportune also the inclusion of the modules on the profiler and their evaluation in a real environment. This would allow the test of the vehicle's model used and its further tuning in order to have a better mathematical representation of its dynamics. Furthermore, it would allow the appraisal of the behavior of the system when in contact with several difficulties that are present in a real environment.

A suggestion to improve the work done is to add new types of lasers, correspondent detection, and extraction of relevant information using them. This would enrich the overall system, providing other types of information other than the altitude. For example, information related to the leaning of the sea floor could be used to improve the control phase of the vehicle.

Since the sensor module was developed in order to be able to be used in other underwater applications, it would be valuable to test the range of application that could benefit from its use. For example, it would be interesting to test the performance of the sensor module when used in an object detection system.

Other suggestion is the use of another sensor in order to acquire the altitude value and its insertion into the overall control system. For example, it would be interesting to compare the performance of the control phase with a real sonar device in a real environment.

Another improvement that could be done is the addition of software and hardware capable of detecting the movement of objects in the range of the camera view. This could improve the performance of the controller phase in cases where an object or particle move in front of the camera view, impacting negatively the measurements obtained.

Appendix A

Attachments

A.1 Extended abstract submitted to the OCEANS 2018 Charleston conference

Altitude control of an underwater vehicle based on computer vision - Extended Abstract

Pedro M. Rodrigues, Nuno A. Cruz, Andry M. Pinto
Faculty of Engineering, University of Porto
up201306231@fe.up.pt, nacruz@fe.up.pt, andry.pinto@fe.up.pt

I. Introduction

The ocean can be seen as Earths central engine when it comes to maintaining its balance. Not only it contains the vast majority of Earths biodiversity and an extreme strategic importance to the security of a nation but it also holds tremendous amounts of resources that enrich the daily life of humankind. In order to fully take advantage of the richness that oceans posses, we need a deeper understanding of the properties and state of those ecosystems.

One of the most relevant ways to gather information from the ocean is through underwater search and exploration missions. It is usual that this type of underwater tasks involves hostile environments far too hazardous for humans, making the human intervention, not ideal in some cases. In order to overcome the challenges that the underwater missions bring, a variety of systems have been developed over the years. The most commonly used solutions are systems based on Remotely Operated Vehicles (ROVs) and/or Autonomous Underwater Vehicles (AUVs).

During the extraction of information, the position control of the vehicle is critical. Specifically, the distance between the vehicle and the sea floor must be warily controlled to ensure its safety and the reliability of the missions that require proximity to the object of interest. Commonly, to deal with the altitude control, a system based on sonar technology is used. Although this solution simplifies the problem and is effective in most cases, it carries a lot of disadvantages in some underwater conditions and in some vehicles with certain specifications. Particularly the sensors based on acoustic waves, like the sonar, might present difficulties on the interpretation of the signals received when the vehicle is too close to the obstacle, requiring a minimum distance to retrieve valuable and reliable information. Furthermore, the inclusion of the sonar sensor demands an increase on the energetic cost of the system that in the case of vehicles powered by an external source through a cable like the ROVs is not a problem, but in AUVs, it might be valuable to avoid it since these vehicles are powered by batteries. Lastly, sometimes the space occupation of the sonar sensor represents a problem in some vehicles with meticulous limits relative to space usage, a common problem found in AUVs.

In order to overcome these problems, the acquirement of the distance measurement can be accomplish through image processing using a system based on a camera and laser pointer devices. Since several underwater vehicles already have an embedded camera and it is common the existence of laser pointer devices as a scale, this approach is opportune and can accomplish the task with high reliability and efficiency.

In this paper we present the development of a system capable of controlling the altitude of an underwater vehicle using computer vision. To provide this functionality, it is required that the system gathers information about the altitude of the vehicle in real-time based on capturing and processing of images containing known marks imposed by laser devices. Through this information, the system must be capable of holding an ordered constant altitude above the sea floor during the progression of its mission.

II. Related work

In the final paper, we will provide an overview of the technologies and solutions that were considered and explored in the development of this project.

III. Problem definition and Solution

To achieve a system capable of controlling the altitude of an underwater vehicle using sensing based on computer vision, it is required the development of a module capable of measuring the distance (Sensor Module) and a module able to filter the data gathered by the sensor module and capable of using this data to control the distance of the vehicle (Control Module).

In order to develop the sensor module, several solutions were researched and analyzed based on the performance of the method, computational demand, and complexity. Based on this analysis, it was decided that the sensor module would be based on two laser pointer devices placed parallel to one another beside a CCD camera and the technology used in order to acquire the distance would be based on simple laser triangulation between the camera and the obstacle. Even though in this project both modules are used together in order to accomplish the goals presented, it is intended that the sensor module represents a system totally independent capable of being used for several other applications. The sensor module must be able to accomplish the following main requirements:

- Allow a vast diversity of components, different camera and laser devices, and also be flexible towards the number of laser devices used;
- Needs to be able to provide information about the quality of the measurements acquired;

- The user must be able to dictate the information provided by the sensor module.

In this project, the underwater vehicle to be used is a profiler with approximately 1.35 m of length and a mass of 11.3 kg. This vehicle main purpose is the navigation in the vertical axis towards the sea floor. Since this vehicle's available space is limited, especially in its extremity, some limitations are imposed in the size of the sensor module components. Specifically, the structure containing the laser pointer devices and the camera must have less than 12cm of diameter.

The solution adopted regarding the control module was chosen comparing the performance of different control approaches, considering their complexity and computational demand. It was decided to use a Kalman Filter in order to filter the information received from the sensor and a PID controller that receives the data produced by the Kalman filter and is capable of controlling the actuation on the motors in order to reach the reference. Unlike the sensor module, the control module needs to be designed based on the vehicle characteristics. The control module needs to verify the following main requirements:

- The Kalman Filter must be able to filter any outlier measurement gathered by the sensor module and guarantee that the control is viable even when no measurement was acquired;
- The Kalman Filter must use the information of the quality of the measurements in order to adjust the impact of the new measurement on the state of the system;
- The PID controller must be flexible allowing the its calibration based on the necessities of the mission.

IV. Implementation

A. Sensor Module

1) Components: The distance measurement infrastructure consists on a Mako G-125 camera with Ethernet communication and 2 green laser pointer devices.

2) Architecture and Implementation: The image processing method developed capable of calculating the distance towards an object based on the acquisition of images through the camera on board of the vehicle is divided into 3 independent phases: detection of the laser dots on the image; calculation of the moments and posterior "centroid" of the laser dots; triangulation method in order to acquire the distance. Therefore, it is possible to calibrate the software to detect different types of lasers. Furthermore, the software allows the choice of what type of operations and functionalities are desired. Operations like moving average, simple average, calculation of independent measures based on the number of laser points and calculation of the quality of those measure are available.

3) Characterization and Tests: In order to characterize and test the sensor module developed, it is firstly necessary to know the limitation of the technology used. Through the triangulation method it is known that the accuracy of the measurements is limited and this dependency is proportional to the distance of the obstacle. In the final paper, an in depth explanation and data referent to this inherent error is going to be presented.

It is also necessary to know the error induced through the utilization of non ideal components. In order to characterize these errors a methodology of testing is used to ensure that it is possible to not only calibrate the sensor module but also examine the possible error present on the measures. In the final paper, the results of these tests and corresponding explanation are going to be presented.

B. Control Module

1) Design of the Control Module: The design approach behind the PID controller development was essentially focused in the creation of a method capable of calibrating the PID controller based on the necessity of the vehicle and its missions. Therefore, based on an already existent mathematical model of the profiler, its state space was calculated in order to acquire the discrete transfer function of the vehicle. For the computer implementation the calculations regarding the PID were based on the digital PID transfer function. The closed loop transfer function of the system was calculated and utilizing the information provided by the root locus and temporal response, it was designed a methodology capable of calibrating the PID parameters relative to the vehicle and mission properties and demands.

2) Simulation and Validation of the PID controller: In order to simulate and validate the PID controller, its step response is analyzed allowing the validation of the demands imposed towards the control of the vehicle. These tests were adapted based on the characterization of the sensor and therefore, an error and time delay relative to the sensor module on the acquisition of the measures is considered.

Furthermore, the control law and the vehicle mathematical model are tested utilizing the UWsim simulation environment in order to have a better visual perspective of the actuation and movement of the vehicle.

V. Integration and operational tests

The test of the behavior of the full system are performed using the UWsim simulation environment and utilizing the ROS communication in order to integrate both modules. The results acquired during these tests will be presented in the final paper.

In order to validate the systems created in a real environment, a series of operational tests are performed where the profiler is commanded to different altitudes. These tests are realized on a tank where the environment conditions are controllable and the results can be compared to the exact values. In the final paper, we will provide the results of these experiments and an in depth analysis of those results.

VI. Conclusion and future work

The final section of the paper will discuss the results obtained and the future work.

References

[1] J. M. Monteiro and N. A. Cruz. Development of an autonomous underwater profiler for coastal areas. In *OCEANS 2017 8211; Anchorage*, pages 1–8.

[2] C. Cain and A. Leonessa. Laser based rangefinder for underwater applications. In *2012 American Control Conference (ACC)*, pages 6190–6195. doi:10.1109/ACC.2012.6315182.

[3] A. Waltz. An improved light source for underwater illumination. In *Ocean 73 - IEEE International Conference on Engineering in the Ocean Environment*, pages 4–6. doi:10.1109/OCEANS.1973.1161211.

[4] Chen Hsin-Hung and Lee Chia-Ju. A simple underwater video system for laser tracking. In *OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No.00CH37158)*, volume 3, pages 1543–1548 vol.3. doi:10.1109/OCEANS.2000.882160.

[5] Thor Fossen. *A Guidance and Control of Ocean Vehicles*. 08 1994.

[6] C. N. McLean, J. E. Manley, and F. Gorell. Ocean exploration: building innovative partnerships in the spirit of discovery. In *Proceedings of the 2004 International Symposium on Underwater Technology (IEEE Cat. No.04EX869)*, pages 3–6. doi:10.1109/UT.2004.1405452.

[7] H. Kondo and T. Ura. Navigation of an auv for investigation of underwater structures. *Control Engineering Practice*, 12(12 SPEC. ISS.):1551–1559, 2004. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-3242886779&DOI=10.1016%2fj.conengprac.2003.12.005&partnerID=40&md5=6116ea3069710364a8bad3fc366f6db1, doi:10.1016/j.conengprac.2003.12.005.

[8] B. K. Sahu and B. Subudhi. The state of art of autonomous underwater vehicles in current and future decades. In *2014 First International Conference on Automation, Control, Energy and Systems (ACES)*, pages 1–6. doi:10.1109/ACES.2014.6808014.

[9] C. Mai, S. Pedersen, L. Hansen, K. L. Jepsen, and Yang Zhenyu. Subsea infrastructure inspection: A review study. In *2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS)*, pages 71–76. doi:10.1109/USYS.2016.7893928.

[10] R. Bogue. Underwater robots: A review of technologies and applications. *Industrial Robot*, 42(3):186–191, 2015. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84930238996&DOI=10.1108%2fIR-01-2015-0010&

partnerID=40&md5=3cfcc8ca261e72158bd2d2028027d3b3, doi:10.1108/IR-01-2015-0010.

[11] K. Poore, C. Kitts, G. Wheat, and W. Kirkwood. A small scale rov for shallow-water science operations. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–6. doi:10.1109/OCEANS.2016.7761405.

[12] H. Kondo, T. Ura, Y. Nose, J. Akizono, and H. Sakai. Visual investigation of underwater structures by the auv and sea trials. In *Oceans 2003. Celebrating the Past ... Teaming Toward the Future (IEEE Cat. No.03CH37492)*, volume 1, pages 340–345 Vol.1. doi:10.1109/OCEANS.2003.178581.

[13] N. Hansen, M. C. Nielsen, D. J. Christensen, and M. Blanke. Short-range sensor for underwater robot navigation using line-lasers and vision. *IFAC-PapersOnLine*, 28(16):113–120, 2015. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84992525104&doi=10.1016%2fj.ifacol.2015.10.267&partnerID=40&md5=a3fe12432a5e7e5f40b51f863551391e, doi:10.1016/j.ifacol.2015.10.267.

[14] C. Becker, D. Ribas, and P. Ridao. Simultaneous sonar beacon localization and auv navigation. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 9, pages 200–205. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84900489794&DOI=10.3182%2f20120919-3-IT-2046.00034&partnerID=40&md5=d81f33a4f0e02a94092301ed638fb289, doi:10.3182/20120919-3-IT-2046.00034.

[15] Joe Cuschieri and Shahriar Negahdaripour. Use of forward scan sonar images for positioning and navigation by an auv. In *Oceans Conference Record (IEEE)*, volume 2, pages 752–756. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0032279122&partnerID=40&md5=e0bd267c200edea6c7034e0bc678b4a0.

[16] S. Zhao, T. F. Lu, and A. Anvar. Automatic object detection for auv navigation using imaging sonar within confined environments. In *2009 4th IEEE Conference on Industrial Electronics and Applications, ICIEA 2009*, pages 3648–3653. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-70349311759&DOI=10.1109%2fICIEA.2009.5138887&partnerID=40&md5=94a98dea286f743366344ff55548f1a4, doi:10.1109/ICIEA.2009.5138887.

[17] L. Paull, S. Saeedi, M. Seto, and H. Li. Auv navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, 39(1):131–149, 2014. doi:10.1109/JOE.2013.2278891.

[18] H. Kondo, T. Maki, T. Ura, Y. Nose, T. Sakamaki, and M. Inaishi. Structure tracing with a ranging system using a sheet laser beam. In *Proceedings of the 2004 International Symposium on Underwater Technology (IEEE Cat. No.04EX869)*, pages 83–88. doi:10.1109/UT.2004.1405483.

[19] S. Zhigang, M. Xiaochuan, L. Yu, and Y. Shefeng. Attitude determination of autonomous underwater vehicles based on pressure sensor array. In *2015 34th Chinese Control Conference (CCC)*, pages 5517–5520. doi:10.1109/ChiCC.2015.7260501.

[20] F. Takemura, R. Taba, K. Hirayama, S. Tansuriyavong, K. Kawabata, S. Sagara, and K. Ogasawara. Development of an altitude-keeping system for underwater robots using laser beams. *Artificial Life and Robotics*, 22(4):405–411, 2017. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85026902493&DOI=10.1007%2fs10015-017-0377-y&partnerID=40&md5=50bb937c4ca364bbd4007ada16d19354, doi:10.1007/s10015-017-0377-y.

[21] A. Cunningham. Design of a ccd camera for space surveillance. In *2016 IEEE Aerospace Conference*, pages 1–9. doi:10.1109/AERO.2016.7500521.

[22] R. D. LaBelle and S. D. Garvey. Introduction to high performance ccd cameras. In *Instrumentation in Aerospace Simulation Facilities, 1995. ICIASF '95 Record., International Congress on*, pages 30/1–30/5. doi:10.1109/ICIASF.1995.519136.

[23] Z. Chu, D. Zhu, and C. Luo. Adaptive neural sliding mode trajectory tracking control for autonomous underwater vehicle without thrust model. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pages 1639–1644. doi:10.1109/COASE.2017.8256339.

[24] A. Hanai, S. K. Choi, and J. Yuh. A new approach to a laser ranger for underwater robots. In *IEEE International Conference on Intelligent Robots and Systems*, volume 1, pages 824–829. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0346149784&partnerID=40&md5=969df58b2e599bcd97634d08cc718e25.

[25] P. Corke. Image processing. In *Springer Tracts in Advanced Robotics*, volume 73, pages 285–333. 2011. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85028769956&DOI=10.1007%2f978-3-642-20144-8_12&partnerID=40&md5=8adc1ca7cbd612e72909edd188bad52b, doi:10.1007/978-3-642-20144-8_12.

[26] P. Mukhopadhyay and B. B. Chaudhuri. A survey of hough transform. *Pattern Recognition*, 48(3):993–1010, 2015. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84916877463&DOI=10.1016%2fj.patcog.2014.08.027&partnerID=40&md5=d86dd73f511627b5c3c4f937f74faa5c, doi:10.1016/j.patcog.2014.08.027.

[27] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek. A brief introduction to opencv. In *2012 Proceedings of the 35th International Convention MIPRO*, pages 1725–1730.

[28] M. H. Marzaki, M. H. F. Rahiman, R. Adnan, and M. Tajjudin. Real time performance comparison between pid and fractional order pid controller in smisd plant. In *2015 IEEE 6th Control and System Graduate Research Colloquium (ICSGRC)*, pages 141–145. doi:10.1109/ICSGRC.2015.7412481.

[29] P. Gupta and R. Gupta. Depth control technique for an autonomous underwater vehicle system. In *2016 International Conference on Communication and Signal Processing (ICCSP)*, pages 0019–0022. doi:10.1109/ICCSP.2016.7754217.

[30] V. Upadhyay, S. Gupta, A. C. Dubey, M. J. Rao, P. Siddhartha, V. Gupta, S. George, R. Bobba, R. Sirikonda, A. Maloo, and V. G. Idichandy. Design and motion control of autonomous underwater vehicle, amogh. In *2015 IEEE Underwater Technology (UT)*, pages 1–9. doi:10.1109/UT.2015.7108287.

[31] Paulo H Buscariollo. *Sistema de posicionamento dinâmico baseado em visão computacional e laser*. Thesis, 2008.

[32] L. Ye, G. Hongda, G. Hao, J. Yanqing, A. Li, and M. Teng. The improved adaptive hybrid fuzzy control of auv horizontal motion. In *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 408–414. doi:10.1109/ICCWAMTIP.2016.8079883.

[33] W. Shikai, J. Hongzhang, and M. Lingwei. Adaptive sliding mode control of auv to optimize the voyage. In *2016 Chinese Control and Decision Conference (CCDC)*, pages 6838–6843. doi:10.1109/CCDC.2016.7532230.

[34] Z. Chu and D. Zhu. Adaptive sliding mode heading control for autonomous underwater vehicle including actuator dynamics. In *OCEANS 2016 - Shanghai*, pages 1–5. doi:10.1109/OCEANSAP.2016.7485568.

[35] G. C. Karras and K. J. Kyriakopoulos. Localization of an underwater vehicle using an imu and a laser-based vision system. In *2007 Mediterranean Conference on Control & Automation*, pages 1–6. doi:10.1109/MED.2007.4433777.

[36] Wallis. Christopher. *An Investigation into Computer Vision Techniques for Underwater Object Recognition*. Thesis, 2006.

[37] N. Y. Ko and T. G. Kim. Comparison of kalman filter and particle filter used for localization of an underwater vehicle. In *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 350–352, Nov 2012. doi:10.1109/URAI.2012.6463013.

[38] L. Fei and Z. Xinying. Underwater target tracking based on particle filter. In *2012 7th International Conference on Computer Science Education (ICCSE)*, pages 36–40, July 2012. doi:10.1109/ICCSE.2012.6295021.

[39] Allied vision - mako g-125. https://www.alliedvision.com/en/products/cameras/detail/Mako%20G/G-125.html. Accessed: 2018-05-23.

[40] Odicforces - green (532nm) laser modules. https://www.bluerobotics.com/store/electronics/bar30-sensor-r1/. Accessed: 2018-05-23.

[41] Andry Pinto, Paulo Costa, and A Moreira. An architecture for visual motion perception of a surveillance-based autonomous robot. In *2014 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2014*, 03 2014.

[42] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[43] Ibrahim A . El-sharif, Fathi O. Hareb, and Amer R. Zerek. Design of discrete-time pid controller. In *International Conference on Control, Engineering Information Technology (CEIT'14) Proceedings - Copyright IPCO-2014 ISSN 2356-5608.*

[44] Bluerobotics - t100 thrusters. https://www.bluerobotics.com/store/thrusters/t100-thruster. Accessed: 2018-05-23.

[45] Afro esc. afro esc 20a user manual. Accessed: 2018-05-23.

[46] J.; Fernandez J.J.; Sanz P.J. Prats, M.; Perez. An open source tool for simulation and supervision of underwater intervention missions. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2577-2582, 7-12 Oct. 2012*.

[47] Uwsim virtual underwater scenario tutorial. http://www.irs.uji.es/uwsim/wiki/index.php?title=Configuring_and_creating_scenes. Accessed: 2018-06-01.

[48] Blue robotics bar30. https://odicforce.com/epages/05c54fb6-7778-4d36-adc0-0098b2af7c4e.sf/en_GB/?ObjectPath=/Shops/05c54fb6-7778-4d36-adc0-0098b2af7c4e/Categories/Background_and_Projects/Green532nm_Laser_Modules__12mm_Brass_Case_type. Accessed: 2018-05-23.